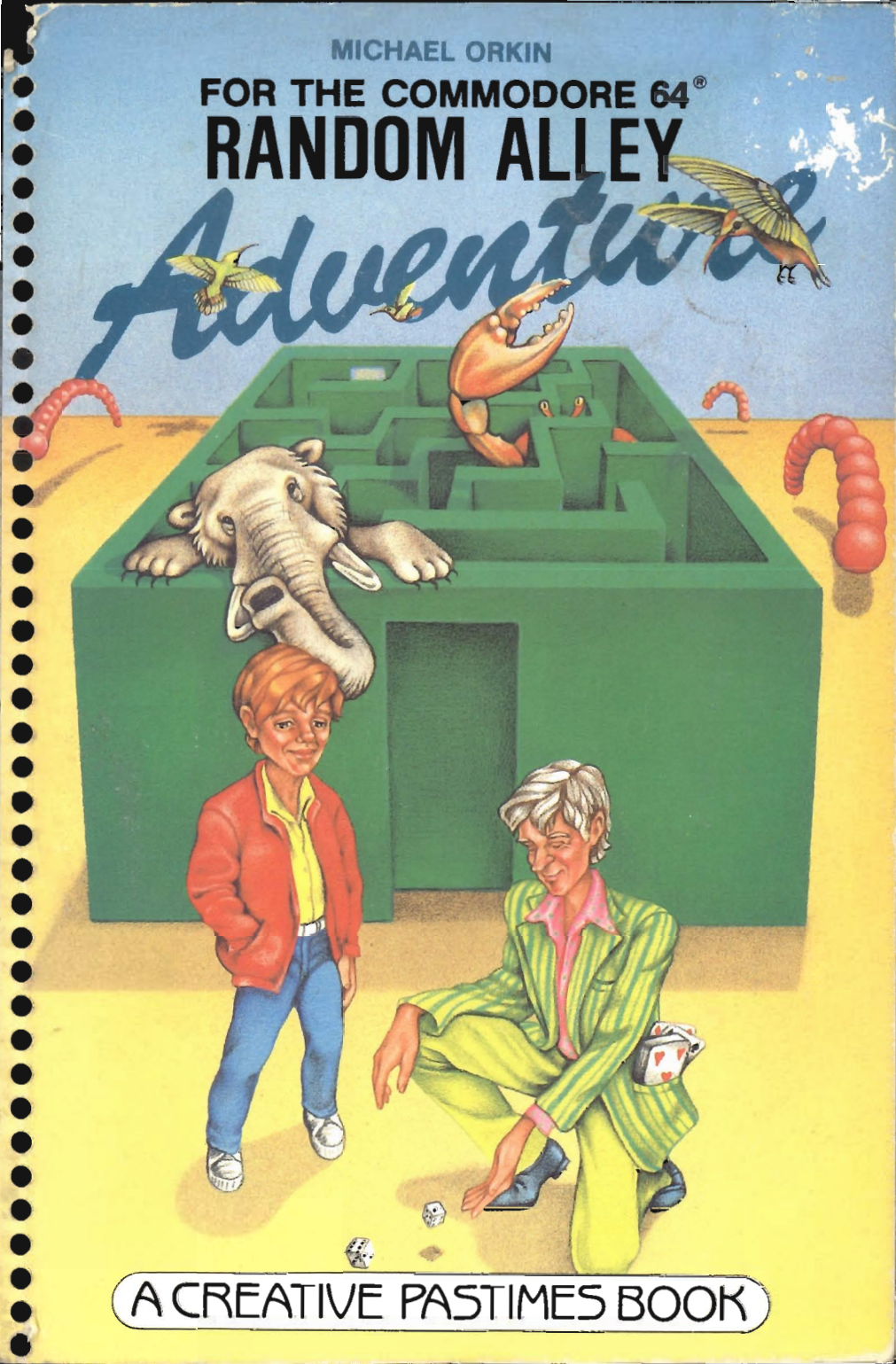
MICHAEL ORKIN

FOR THE COMMODORE 64®
RANDOM ALLEY Adventure



A CREATIVE PASTIMES BOOK

This book belongs to

Random Alley Adventure

for the Commodore 64®

Michael Orkin

Illustrations by Donna Ward

Herbert Kohl, Series Adviser

A Creative Pastimes Book
Reston Publishing Company
A Prentice-Hall Company
Reston, Virginia

Library of Congress Cataloging in Publication Data

Orkin, Michael.
Random Alley adventure for the Commodore 64.

"A Creative Passtimes book."
1. Computer games. 2. Commodore 64 (Computer)—Programming.
3. Basic (Computer program language) I. Title. GV1469.2.076
1984 794.8'2 83-26857
ISBN 0-8359-6407-8

Interior design and production by Laura Cleveland

10 9 8 7 6 5 4 3 2 1

Printed in the United States of America.

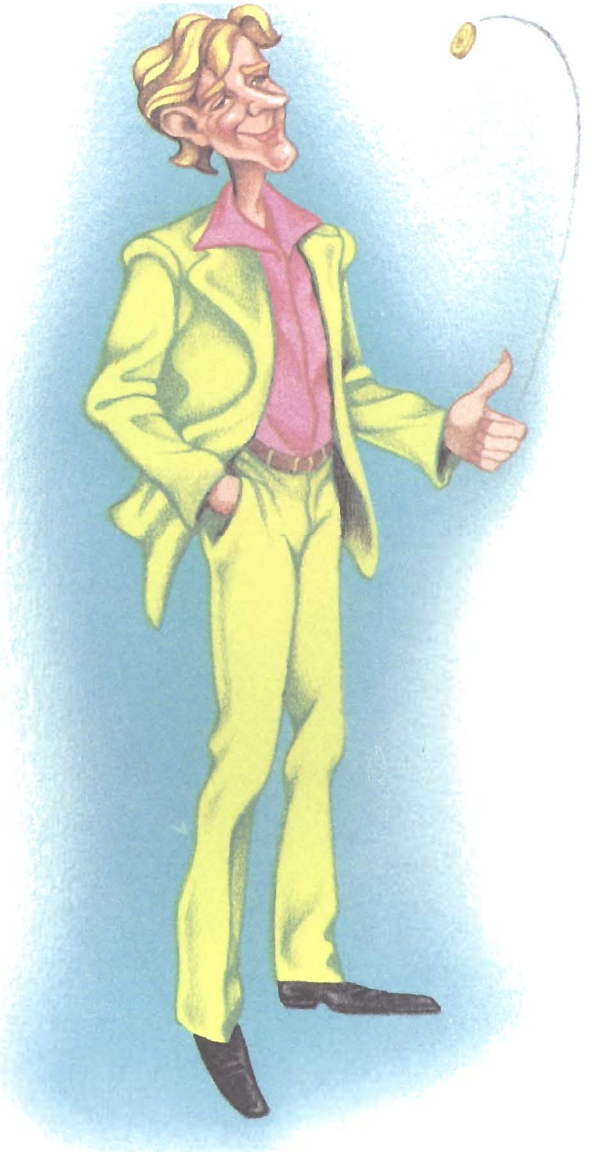
Contents

Preface

Random Alley Adventure is for anyone interested in the laws of chance. It is the story of a teenage boy who finds himself in a world of randomness. During his adventure he encounters gamblers, game players, and video game environments. The book also contains computer programs for the Commodore 64 that illustrate events in the story, and games including craps and roulette.

Because a computer can play a game of chance over and over very quickly, it can be used to demonstrate the law of averages. This fascinating mathematical law states that, in repeated plays of a game of chance, the unpredictable outcomes of individual plays will stabilize to a predictable limiting average. For example, although one can't predict with certainty whether a tossed coin will come up heads or tails, in repeated tosses the coin is sure to come up heads approximately 50% of the time and tails 50% of the time. It proves that gamblers who play unfavorable games will always lose in the end. The programs in the book demonstrate this.

All of the programs are written in the BASIC programming language. You don't have to know anything about programming to use them. The programs and story complement each other to create a simple introduction to randomness.



1

The Thin Man

"Harold Bloomgarden! Come downstairs this minute and pick up your books!" shouted Harold's mother.

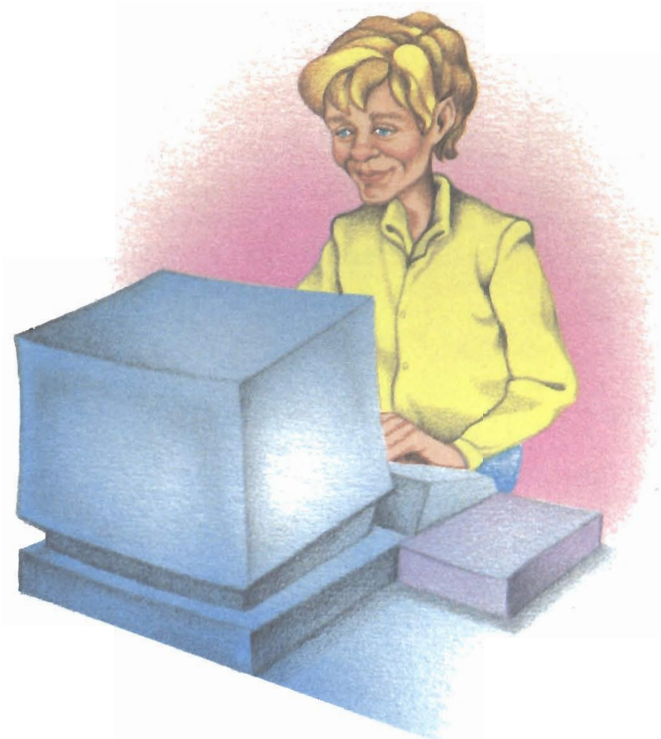
"In a minute," Harold yelled back as he tried to put the finishing touches on a computer program.

"How can I be expected to get any work done when I'm constantly interrupted?" Harold mumbled to himself.

Harold stomped down the stairs and picked up his books. On his way back to his room he decided to go for a bus ride.

Harold liked to ride the bus randomly around town; destinations weren't important. Seeing different places and people helped him relax and think up new ideas.

As the bus rounded a corner near the downtown shopping district, it made an unexpected turn and entered a strange looking alley Harold had never seen before. The street



sign said "Random Alley." All the streets were very narrow like alleys. The buildings were small and painted weird colors, but that could have been the light, which seemed artificial.

The only other passenger on the bus was a thin man with scraggly hair. Harold could not remember seeing the thin man get on the bus. As the bus neared a stop, the thin man took a quarter out of his pocket. He flipped the coin in the air, caught it, and observed the result. Then he put the quarter back in his pocket and casually looked out the window. At the next stop the same thing happened.

At the third stop, the thin man tossed the coin again. "Heads!" he shouted with a smile. He quickly got up and left the bus. Harold got up too and trailed down the steps of the bus after him.

The thin man walked up the street in the direction from which the bus had come. Harold had to run to catch up with him. "Hello!" said Harold when he caught up with the thin man. "My name is Harold. What's yours?" Not getting a response, Harold added nervously, "I've never been in this part of town before. Could you tell me where I am?"

"Can't talk now," said the thin man as he walked briskly up the street. "I'm late for a roulette game. I have good luck if I wait until my coin lands heads before getting off the bus. At each stop, starting where I want to get off, I toss the coin. If it comes up heads I get off. If it comes up tails I stay on. Today it came up tails twice, so now I have to walk two extra blocks."

The thin man, with Harold close on his heels, came to an intersection, where the man stopped and took out his coin. "This is a special intersection," he said. "I don't cross the street here unless the coin comes up heads."

The thin man flipped his coin but missed catching it. The quarter landed on the sidewalk and rolled along until it fell into a sewer.

"Oh no!" groaned the thin man. "Now I won't know if I'm supposed to cross the street until somebody loans me a quarter."

"You can use my pocket computer to simulate the toss of a coin," said Harold eagerly as he dug in his pocket for a small computer.

Here is a program that will instruct the computer to print a sequence of random decimal fractions. The program will keep printing these numbers until you stop it by pressing the **RUN/STOP** key.

Numerical information can be stored in the computer's memory by using a numeric "variable," which is a type of label for a memory location. In Commodore BASIC, a numeric variable can be denoted by a capital letter. You can assign a value to a variable by using the *LET statement*. In fact, you don't even have to use the word LET in the statement. In line 10 of the following program, we "let" X equal the value of the random number we generate using RND(1). The *PRINT statement* causes the computer to print information on your TV screen. In line 5, press the **CLR/HOME** key while the **SHIFT** key is pressed. This will cause the computer to clear the screen. Lines 8 and 9 cause the computer to skip a space. The *GOTO statement* in line 40 instructs the computer to go to Line 10. Line 30 in the following program is a *delay statement*. It causes the computer to delay "400 times" before proceeding to the next instruction. Otherwise, the random numbers would be printed too fast to read. Now enter and RUN the following program. Don't forget to press **RETURN** each time you type a line and to press the **RUN/STOP** key when you want the program to stop. After you have run the program a few times, try varying the number 400 in the delay statement (line 30). Another way to slow down the printing is by pressing the **CTRL** key.

```
 5 PRINT "SHIFT CLR/HOME"
 7 PRINT "RANDOM DECIMAL FRACTIONS"
 8 PRINT
 9 PRINT
10 X=RND(1)
20 PRINT X
30 FOR D=1 TO 400:NEXT D
40 GOTO 10
```

You can eliminate the fractional part of a number by using the INT function. INT(X) drops the fractional part of X. For example, INT(3.1416)=3 and INT(.3679)=0. Since the command RND(1)

generates a random number between 0 and 1, multiplying by 2 gives a random number between 0 and 2. In BASIC, the asterisk is the symbol for multiplication, so 2*RND(1) will be a random number between 0 and 2. You can change the random number 2*RND(1) into a "random digit," 0 or 1, by using the INT function to drop its fractional part. There is a 50–50 chance that this random digit will be 0 and a 50–50 chance that it will be 1. This is just like tossing a coin, in which there is a 50–50 chance that heads will come up and a 50–50 chance that tails will come up. Enter and RUN the following program, which will have the computer print a sequence of random digits. Since these digits are either 0 or 1, you can call them *random "binary" digits.* (*Binary* means two valued, like 0 or 1.)

```
 5 PRINT "SHIFT CLR/HOME"
 7 PRINT "RANDOM BINARY DIGITS"
 8 PRINT
 9 PRINT
10 X=INT (2*RND(1))
20 PRINT X;
30 FOR D= 1 TO 400:NEXT D
40 GOTO 10
```

You can make a few modifications in the above program and have the computer designate 0 as "heads," and 1 as "tails." Enter and RUN the following program, in which the computer will interpret random binary digits as the results of tossing a coin.

```
 5 PRINT "SHIFT CLR/HOME"
 7 PRINT "COIN TOSSING"
 8 PRINT
 9 PRINT
10 X=INT(2*RND(1))
20 IF X=0 THEN PRINT "HEADS"
25 IF X=1 THEN PRINT "TAILS"
40 FOR D=1 TO 400:NEXT D
50 GOTO 10
```

The thin man eyed Harold's pocket computer suspiciously but, when the computer's display screen read "Heads" after it simulated the coin toss, the thin man smiled and crossed the street.

"I heard about somebody who spent 30 years standing on a corner, tossing a coin over and over, getting tails every time," said the thin man. "Talk about bad luck!"

Harold rolled his eyes. "He must have had a coin with two tails," he said, politely. Harold knew a lot about randomness but he didn't want to show off. "You can be lucky or unlucky for a while but then things become very predictable," he said. "Since there's a 50–50 chance of getting heads or tails on each toss, eventually both will come up about 50% of the time. I'll show you what I mean."

Although the computer can't actually toss a coin, it can generate random numbers that have the same mathematical properties as coin-tossing results. This process is called *simulation*. One thing a computer can do that a person can't do is simulate an experiment like coin tossing and perform the experiment over and over very quickly, keeping an accurate record of the results. This will allow us to observe a very interesting phenomenon known as the *law of averages*. The law of averages says that in many trials of a random experiment, the actual fraction of times a particular event occurs approaches the probability of the event. When applied to coin tossing, the law of averages says that if you toss a coin a large number of times, heads will come up about half the time and tails will come up about half the time, no matter what. This is bad news for gamblers, who hope for a run of "luck" in an unfavorable game. This cannot happen in the long run.

You can use the computer to demonstrate the law of averages. In fact, using a computer is probably the best way to do this. In the following program, the computer will toss a coin a specified number of times and print out the fraction of times that heads and tails come up. An *INPUT statement* will be used to allow you to enter the number of tosses you want. When the computer encounters an INPUT statement it will print a question mark on your TV screen.

After you input the number of tosses you want, press the RETURN key and the computer will proceed to the next instruction in the program.

We will use a *FOR/NEXT loop* to toss the coin the specified number of times. The loop performs every instruction between the FOR and NEXT statements as many times as are specified. Keeping track of the total number of heads and tails is done with the variables H and T. Each time heads occurs, 1 is added to H, and each time tails occurs, 1 is added to T. The total number of tosses is Y. At the end we will obtain the fraction of heads and tails by dividing H and T by Y. (The division symbol is /.) An indicator on the screen will show you how fast the computer is doing its simulation.

When a comma is used between two or more items in a PRINT statement (lines 28, 62), the computer skips to the next tab position between items. The automatic tab settings are every 10 spaces across the line. Other tab settings can be set in a program with the *TAB statement*.

In 1,000 or more tosses, the fraction of heads and tails will most likely be very close to .5. Type NEW and press RETURN to clear the computer's memory. Then enter and RUN the program. Experiment by inputting different values. To exit from the program it may be necessary to press the RESTORE key while the RUN/STOP key is pressed. In line 66, press the ⇑ CRSR ⇓ key while the SHIFT key is pressed. This will cause the cursor to jump up a line, so that it will keep the tally in the same place.

```
 5 PRINT "SHIFT CLR HOME";
10 PRINT "COIN TOSSING SIMULATION"
15 PRINT "HOW MANY TOSSES";
20 INPUT Y
22 PRINT
28 PRINT "HEADS", "TAILS"
30 FOR J=1 TO Y
40 X=INT(2*RND(1))
50 IF X=0 THEN H=H+1
60 IF X=1 THEN T=T+1
62 PRINT H, T
66 PRINT "CRSR ⇑";
70 NEXT J
80 PRINT
```

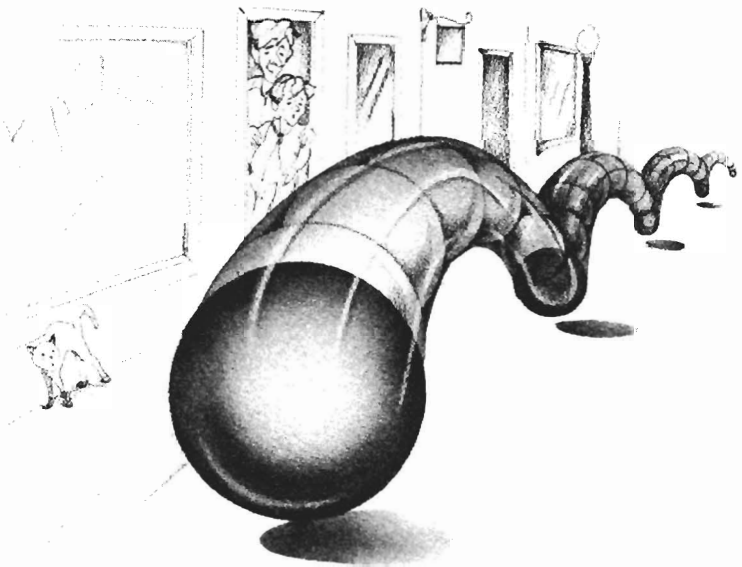
```
 85 PRINT
110 PRINT "FRACTION OF HEADS = ";H/Y
120 PRINT "FRACTION OF TAILS = ";T/Y
```

"Very interesting," said the thin man, "but I really don't have time for such things." He started walking quickly down the street again.

The thin man and Harold had gone a short distance when the thin man suddenly shouted, "WATCH OUT!" He grabbed Harold and pulled him into the doorway of a shop just in time to avoid a bouncing giant rubber ball about the size of Harold's Uncle Phil.

"What was that?" gasped Harold after the giant ball had bounced out of sight.

"It was a bouncing giant rubber ball," said the thin man dryly. "They bounce around at random. If you're unlucky, you get crushed. You get used to them after a while. Are you new around here?"



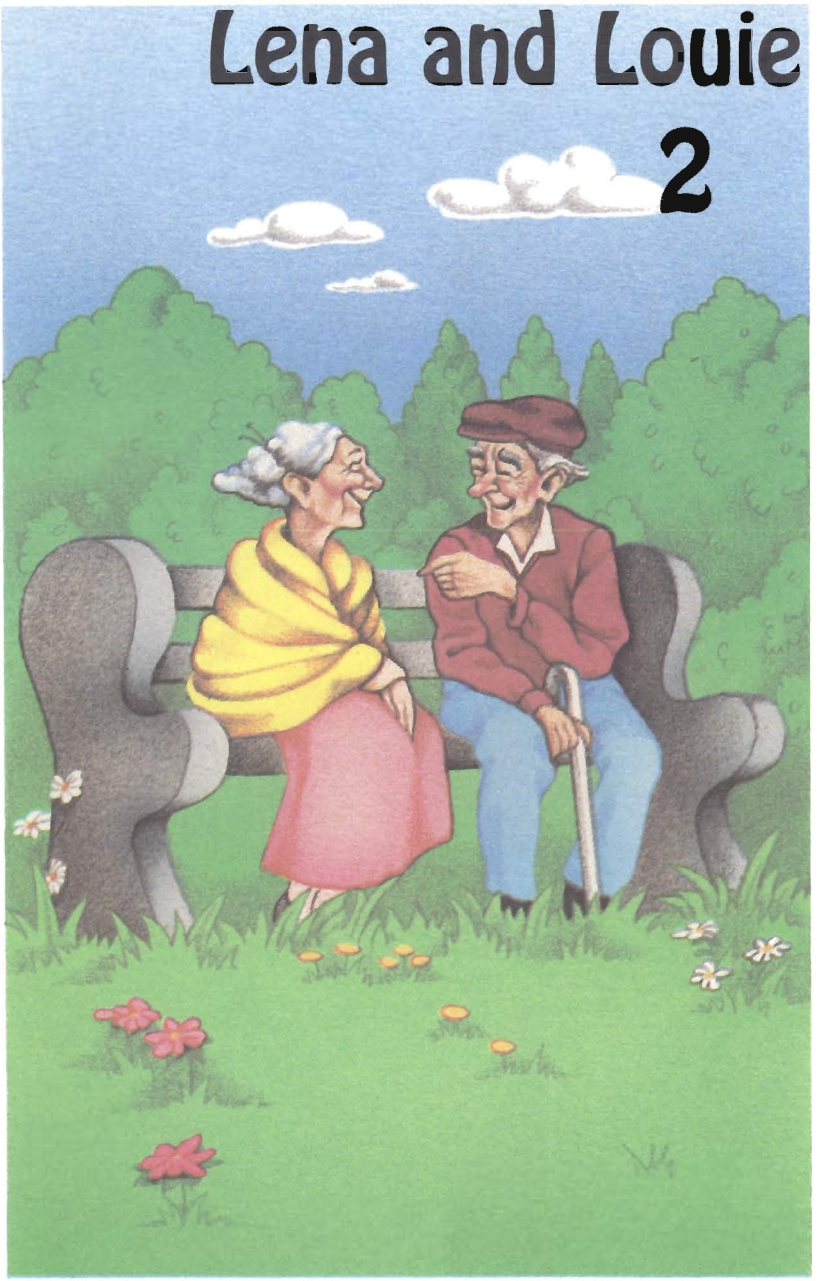
"I'm from a different part of town," said Harold. "I was riding the bus and all of a sudden we went through a place called Random Alley."

"If I were you I'd get out while I could," said the thin man. "After you've been here a while, it's very difficult to leave. Especially if the Controller of Chance finds you."

"Who's the Controller of Chance?" asked Harold.

"He runs this place," said the thin man. "He has magic powers and can control chance events. He likes people to stay here and play games." The thin man stopped in front of a gate along a fence that bordered a vacant lot between two buildings.

"Well, here I am," he said. "Take my advice and go back where you came from." The thin man opened the gate and disappeared behind the fence.


Lena and Louie
2

Harold tried to open the gate to follow, but it locked when the thin man closed it. Desperately, Harold shook the gate but nothing happened. Harold wandered aimlessly down the street, trying to decide what to do. At the end of the block, he came to a large, grassy park dotted with trees.

An elderly couple was sitting on a bench. As Harold got closer he saw that they were playing a game.

Harold cleared his throat nervously. "I really didn't mean to eavesdrop," he apologized. "But are you playing a strategy game?"

"It's a guessing game," the woman, whose name was Lena, smiled kindly at Harold. "The defender tries to guess how the invader will attack. I'm the defender and Louie is the invader. At the count of three, we each can say AIR or LAND. If I say the same thing as Louie, then I have guessed his attack strategy and I win. If we say different things—that is, if one of us says AIR and the other says LAND—he has avoided my defense and he wins. Louie just can't seem to outsmart me, though. What's the score now, dear?" Lena said sweetly to Louie.

You can write a program for the AIR-LAND invasion game in which the computer is the invader and you are the defender. Remember the rules: If you correctly guess the invader's route (AIR or LAND), you win. If your guess is incorrect, you lose. The computer will pick an invasion route at random. It will invade by air if the random digit INT(2*RND(1)) equals 0 and by land if INT(2*RND(1)) equals 1. Written information is stored by the computer as a "string variable," which can be denoted by a letter followed by a dollar sign (like A$). String variables are stored differently in the computer's memory than are numeric variables because you don't do arithmetic with them. The invasion route will be stored as the string variable B$. You can use an INPUT statement to allow you to pick a defense. The computer will then compare your choice with its choice and announce the result. The "inequality" symbols ⟨ (less than) and ⟩ (greater than), when used together as ⟨⟩, mean "not equal to." The *IF/THEN statement* will cause the

computer to branch to the appropriate part of the program if the invasion route is not equal to the defense strategy. If a line in the program is longer than the line on the screen, it will automatically continue on the next line of the screen.

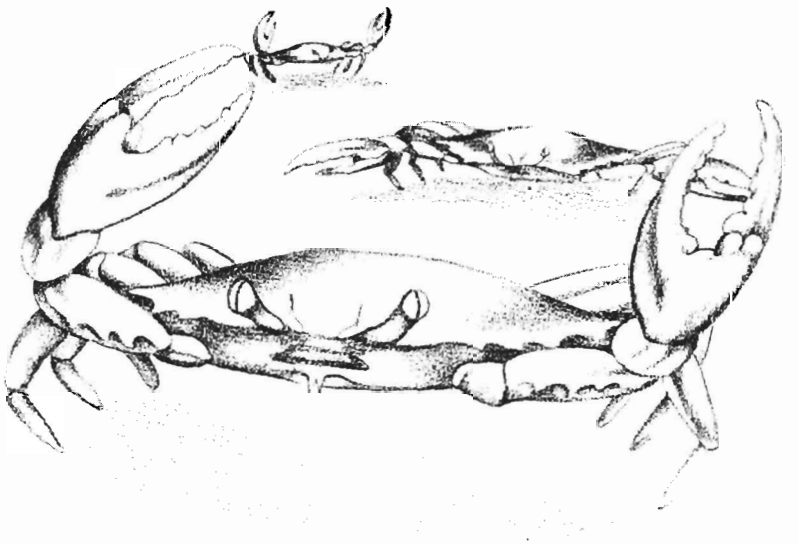
Clear the computer's memory by typing NEW and pressing the **RETURN** key. Then RUN the program to play the game. To exit the program, press the **RESTORE** key while the **RUN/STOP** key is also pressed.

```
 10 PRINT "SHIFT CLR/HOME"
 15 PRINT "AIR-LAND INVASION"
 20 PRINT
 25 X= INT(2*RND(1))
 30 IF X = 0 THEN B$ = "AIR"
 40 IF X = 1 THEN B$ = "LAND"
 50 PRINT "INPUT DEFENSE (AIR OR LAND)"
 60 INPUT A$
 65 IF A$ <> "AIR" THEN IF A$ <> "LAND" THEN GOTO 50
 70 PRINT
 80 PRINT "I CHOSE TO INVADE BY "; B$
 90 PRINT
100 IF B$ = A$ THEN PRINT "YOU WIN"
110 IF B$ <> A$ THEN PRINT "I WIN!!"
120 PRINT
130 GOTO 25
```

Harold and Lena turned expectantly to Louie to hear the score. "CRABS!" he shouted, and jumped up on a table next to their bench. Lena jumped on the table also. Harold was still trying to figure out what "CRABS" meant for a score when Lena grabbed his hand.

"Quick—up here with us!" shouted Lena. Harold, who was more than a bit confused, jumped onto the table just in time to avoid the charge of a group of enormous crabs who went scuttling quickly past. The crabs had pincers large enough to snap Harold in two. One of the crabs stopped next to the table and eyed Harold hungrily.

"What's going on here?" gasped the terrified boy.



"Oh, it's just the monster crabs," said Louie, matter-of-factly. "They charge through the park from time to time. It's easy to avoid them if you know what to do. They never leave the grass," he added. "See that gray tree over there?" Louie pointed to a large, gray tree. "The tree has pretty purple flowers whose smell will make the crabs run away. You have to climb the tree and pick the flowers when you need them because their smell only lasts a few minutes after they're picked. I'm too old to climb, so when I'm in the park I stay near a table." Louie grinned like this was the most reasonable statement in the world.

"Giant balls and monster crabs," said Harold. "I think I want to go home."

"Oh, you saw the bouncing balls?" said Lena brightly. Harold wondered if she ever stopped smiling. "Don't worry about them. They never leave the sidewalk."

The monster crab that had stopped by the table seemed to grow tired of watching Harold. With a disappointed hiss it slowly crawled away. When the crab was gone, Harold and Lena and Louie got down from the table.

"Let's resume where we left off," said Lena to Louie. "I think the score was . . ."

"You're ahead 83,950 to 43," said Louie.

Harold thanked Lena and Louie politely for saving his life, but said that he had to go home. He turned to leave, but the street seemed to be going in a different direction than when he entered the park. Also, some of the buildings looked different. Harold turned to Lena and Louie. "That's weird," he said. "The street looks different now."

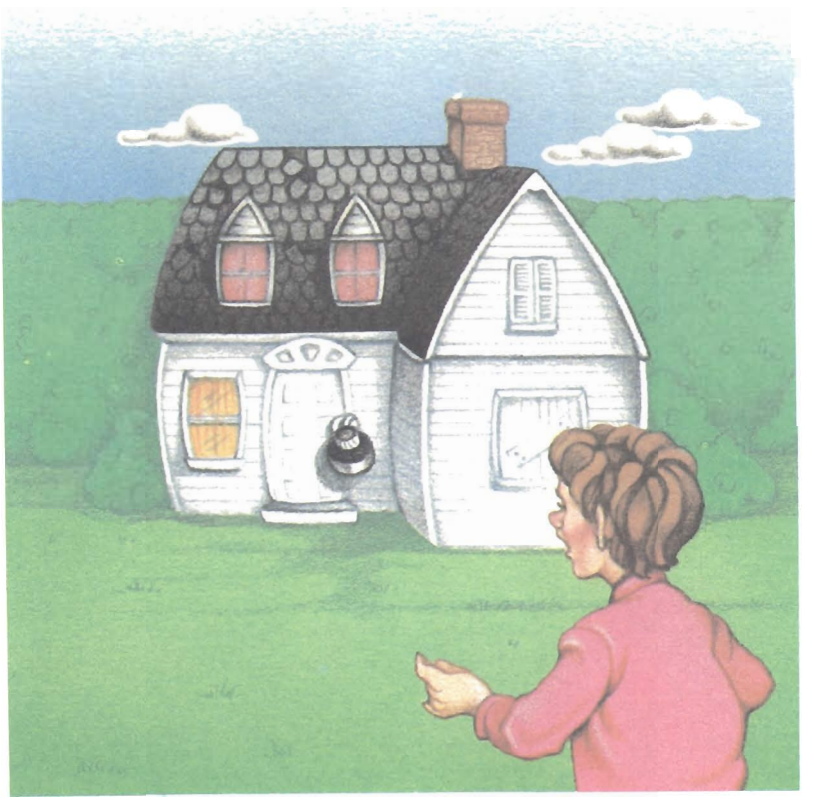
"Oh, yes," said Lena. "Some of the streets change from time to time. It's the work of the Controller of Chance. He says life is more interesting this way. You have to learn your way around." Lena then pointed across the park. "On the other side of the park is a maze," she said. "If you go through the maze, the streets will be the same as when you entered the park."

"Watch out for the killer hummingbirds," advised Louie. "You can scare them off with branches."

"Great—killer hummingbirds!" said Harold.

Harold thanked Lena and Louie for their help and walked across the park toward the maze.

The House 3



It was now mid-afternoon and Harold suddenly realized he was hungry. As he walked across the park, he saw a clown working at a snack stand. He stopped and ordered a hot dog from the clown.

"Coming right up," said the clown, who handed Harold a pie.

"I ordered a hot dog, not a pie," said Harold.

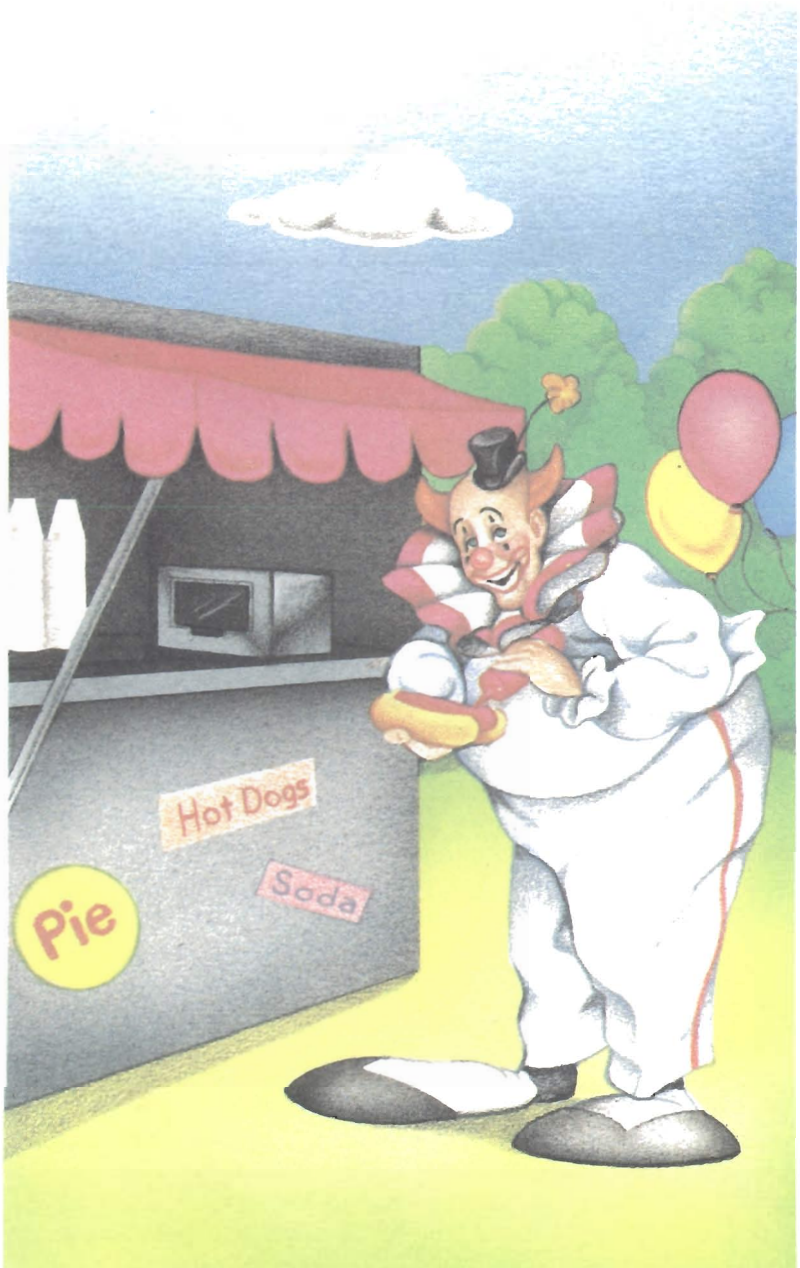
"A hot dog? A pie? What difference does it make?" said the clown. "I have a variety of food. When you order, I pick an item at random and that's what you get. Keep trying, pal. I'm sure you'll get a hot dog eventually."

"No thanks," sighed Harold, who wasn't in the mood for pie and wasn't feeling very hungry anymore.

The random digits 0 and 1 were used to determine the results of tossing a coin. In order to simulate a random experiment that has more than two possible outcomes, the RND(1) function will generate an appropriate number of random digits. The clown in the park has 10 items of food to choose from so, in order to simulate his selection method, you must select a random digit from 1 to 10. First multiply RND(1) by 10 to produce a random number greater than 0 and less than 10. Then use INT to take the integer part of this number to yield a random digit between 0 and 9. Adding 1 gives a random digit between 1 and 10: INT(10*RND(1)) + 1.

The 10 items on the clown's menu will be listed in *DATA statements*. In DATA statements, each unit of information must be separated by a comma. You can have as many DATA statements as you want. They can be anywhere in the program, but it is a good idea to put them at the end.

Data in a program is accessed by a *READ statement*. The READ statement reads data item by item in order through all the DATA statements. Once the data has been exhausted, you cannot read any more unless you put a *RESTORE statement* in the program. The RESTORE statement causes the computer to go back to the first item in the first DATA statement if the program wants more data to be read.


Hot Dogs
Soda
Pie

You can select an item at random from the data list by choosing a digit at random and causing the computer to read that item from the list. In order to get to a particular item, the computer first has to read every item before it. For example, in order to read the eighth item the computer has to first read items one through seven. This can be done by using a *FOR/NEXT loop* which will perform the READ command for the number of times specified. Each time an item is selected at random from the clown's menu in this way a RESTORE statement is used to return to the beginning of the data for the next selection. Type NEW and press the **RETURN** key. Enter and RUN the following program. When you understand how the program works, try adding more items to the clown's menu.

```
 10 PRINT "SHIFT CLR/HOME"
 20 PRINT "SELECTING AN ITEM AT RANDOM"
 25 PRINT
 30 X=INT(10*RND(1))+1
 40 FOR J=1 TO X
 50 READ A$
 60 NEXT J
 70 PRINT A$
 75 FOR D=1 TO 1000:NEXT D
 80 RESTORE
 90 GOTO 30
100 DATA PIE, HOT DOG, ICE CREAM, CANDY
110 DATA HAMBURGER, POPCORN, JUICE
120 DATA FRENCH FRIES, MILK, SALAD
```

Harold continued across the park looking for the maze. At the far side of the park he came to a small, white house. The windows were closed and the curtains were drawn. The paint was peeling and the roof needed repairing. A large combination lock was rusted onto the front door.

Behind the house the park was bordered by a tall hedge. Harold was too short to see over the hedge and it was too thick to wiggle through. He walked along the edge of the hedge

and realized that it bordered the entire end of the park. Leaving the park in this direction would be impossible.

Harold couldn't find the maze and didn't see anyone he could ask for directions. Harold walked back around the little white house. All the windows were tightly closed. There was a back door but it was boarded shut. Harold knocked loudly on the front door but there was no answer. He pulled on the combination lock, but it was attached to a sturdy steel latch.

Harold was about to give up on the house when he noticed a mailbox next to the front door. Hoping he wasn't being rude, he peeked in the mailbox. There he found an envelope with "Lock Combination" written on the outside. Harold opened the envelope and found a pair of dice and a slip of paper with the numbers 3-7-9 written on it.

The excited boy thought 3-7-9 had to be the combination to the lock, but when he tried these numbers the lock failed to open. Harold tried the numbers in backwards order, 9-7-3, but the lock still didn't open. Trying hard not to panic, Harold rolled the dice. The numbers that came up were 1 and 4, for a total of 5. Harold rolled the dice a few more times, observing totals of 6, 8, 7, 10, 9, 7, 3, 6. Harold figured that the dice totals had to hold the answer to the lock combination, but since dice totals are random, he couldn't see how.

Harold stared at the numbers on the paper and thought about dice odds. Since the numbers 1 through 6 are on each die, there are a total of $6 \times 6 = 36$ dice combinations. For example, there are six possible combinations that result in the number "7": 1 on the first die and 6 on the second, 6 on the first die and 1 on the second, 2-5, 5-2, 3-4, 4-3. This means that when you roll a pair of dice, there are 6 chances in 36 or about a 17% chance that "7" will come up. Harold scribbled down a list of dice combinations and percentages on the back of the envelope.

Total	*Combinations*	*Chances*	*Percentage*
2	1-1	1/36	3%
3	1-2 2-1	2/36	6%
4	1-3 3-1 2-2	3/36	8%

Total	*Combinations*	*Chances*	*Percentage*
5	1-4 4-1 2-3 3-2	4/36	11%
6	1-5 5-1 2-4 4-2 3-3	5/36	14%
7	1-6 6-1 2-5 5-2 3-4 4-3	6/36	17%
8	2-6 6-2 3-5 5-3 4-4	5/36	14%
9	3-6 6-3 4-5 5-4	4/36	11%
10	4-6 6-4 5-5	3/36	8%
11	5-6 6-5	2/36	6%
12	6-6	1/36	3%

The numbers 3, 7, and 9 had chances of 6%, 17%, and 11%. Harold was getting excited! He tried the numbers 6, 17, and 11 on the combination lock. When he finished, the lock popped open!

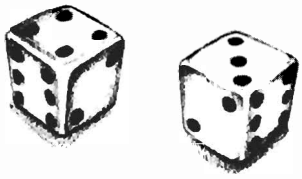
The computer is the perfect tool to simulate dice rolling. A die is a cube with the numbers from 1 to 6 represented by an appropriate number of dots on its six sides. When a die is rolled, each number has an equal (⅙) chance of being face up when the die comes to rest. Actually, this is only true if the die is evenly balanced. There are "loaded" dice for which some sides are more likely to come up than others.

We will write a program that simulates the roll of an evenly-balanced die by using INT(6*RND(1))+1 to generate a random digit between 1 and 6. Type NEW and press the **RETURN** key. Enter and RUN the following program:

```
 5 PRINT "SHIFT CLR/HOME"
 7 PRINT "ROLLS OF A DIE"
 8 PRINT
 9 PRINT
10 X = INT(6*RND(1)) +1
20 PRINT X;
30 FOR D=1 TO 400: NEXT D
40 GOTO 10
```

Most dice games use the sum of the dots on the upturned sides when two dice are rolled. Enter and RUN the following program, which will print the results of rolling a pair of dice:

```
 5 PRINT "SHIFT CLR/HOME"
 7 PRINT "ROLLING A PAIR OF DICE"
 8 PRINT
 9 PRINT
10 X = INT(6*RND(1)) + 1
15 Y = INT(6*RND(1)) + 1
20 PRINT "FIRST DIE = "; X
25 PRINT "SECOND DIE = "; Y
27 PRINT "SUM = "; X+Y
30 FOR D=1 TO 1000: NEXT D
35 PRINT
40 GOTO 10
```



And now for a program that demonstrates the law of averages for dice rolling. Dice probabilities, expressed as percentages, determine the combination of the lock on the door to the white house. By simulating the roll of a pair of dice a large number of times, you can see that the actual results closely fit these percentages.

To keep track of the number of times each result occurs we will use an *array*. An array is an ordered list of numbers. To store an array in the computer's memory, you must first use the "dimension" statement DIM so that the computer will know how many numbers there are in the list. An array name is like a name for a numeric variable. It must begin with a letter and can be followed by a letter or a number. To specify a certain element in the array you give the array name followed by the number of the element in parentheses. The letter A will denote the array that keeps track of dice rolls. A will have a dimension of 12: DIM A(12), with the possible results being tallied in elements A(2)–A(12). For example, if 6 comes up 15 times in 100 rolls, A(6) should equal 15 at the end of the simulation.

To determine the percentage of times a particular result occurs, take the fraction of occurrences, multiply by 100, and take the integer value of the result with .5 added. (The last step rounds off to the nearest whole percent.) Type NEW and press the **RETURN** key.

Now enter and RUN the dice roll simulation program. Use the program for various totals of rolls so that you can see both displays. In 2,000 rolls, you will probably get a close approximation to the dice probabilities.

```
  5 PRINT "[SHIFT] [CLR/HOME]"
  7 PRINT "DICE ROLL SIMULATION"
 10 PRINT "HOW MANY ROLLS";
 20 INPUT T
 25 PRINT
 30 DIM A(12)
 32 FOR S=2 TO 12
 34 A(S)=0
 36 NEXT S
 40 FOR J=1 TO T
 44 PRINT J
 46 PRINT "[CRSR] [↑] ";
 50 X=INT(6*RND(1))+1
 60 Y=INT(6*RND(1))+1
 70 S=X+Y
 80 A(S)=A(S)+1
 90 NEXT J
100 PRINT
110 PRINT "DICE TOTALS"
120 PRINT
130 PRINT " #", "TOT.", "%"
140 PRINT "__ __ __ __ __ __ __ __ __ __ __ __ __ __ __"
150 FOR M=2 TO 12
160 PRINT M, A(M), INT((100*A(M)/T)+.5)
170 NEXT M
```

Harold pushed open the door to the little house and slowly walked inside. It was dark and cool with a musty smell that reminded him of a cave he had once explored. The living room was bare except for an old sofa and a bookcase. Oddly enough, the bookcase was filled with adventure stories and game books.

In the kitchen on a long, wooden table was a bag of peanuts; a jar filled with green liquid; an empty backpack; some change; a rope; a toy horn; a list of registered voters in Cleveland, Ohio, with names, addresses, and birthdates; a small knife; and a flashlight.

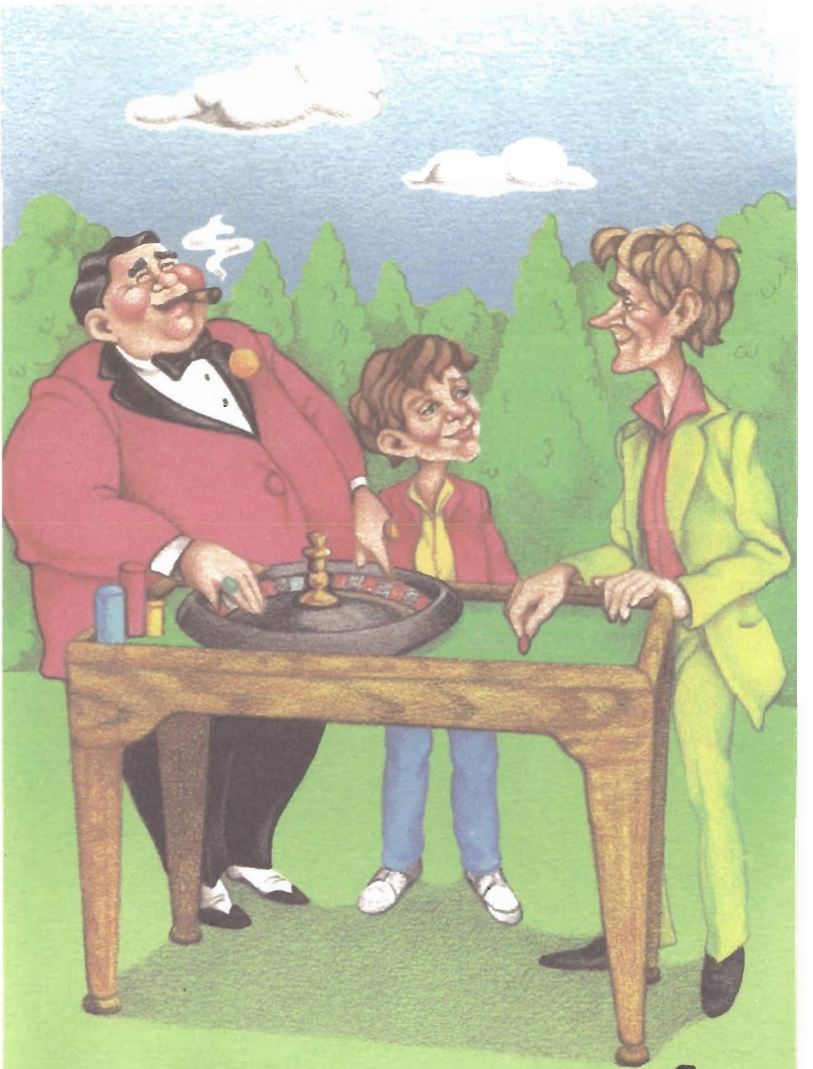
Harold searched all the rooms but couldn't find the maze. He grabbed a handful of peanuts and walked over to the bookcase.

The bookcase and the books were covered with dust. Harold brushed the dust off a few of the books. *Underground Adventures* was the first title he saw. Another was entitled *Survival on Planet X* and another was *The Planet of Lost Games.* Harold started feeling a little better. He loved to read adventure stories. Looking through the books, Harold remembered how hungry he was and began eating the peanuts, tossing the shells in a wastebasket next to the bookcase. When one of the peanut shells missed the wastebasket, Harold bent down

to pick it up. He noticed that a pile of dust next to the bookcase had been disturbed. He also noticed scrape marks on the floor. The marks led to the bookcase. Harold guessed that the bookcase had been moved recently. Curious, he decided to try to move it himself.

Harold pushed hard and the bookcase moved away from the wall. Behind the bookcase was a wooden door about three feet high. Harold opened the door and discovered a narrow staircase that descended into darkness. Forgetting to be afraid, Harold squeezed through the doorway and started to walk down the stairs, but after a couple of steps down it was too dark to see. He remembered the flashlight on the table and ran back up the stairs to get it.

Back in the livingroom, Harold decided he might as well be prepared for whatever he might find so he picked up the backpack and loaded it with everything else on the table; the rest of the peanuts, the toy horn, the green liquid, the knife, the rope, the change, and the list of voters. Then he went back to the hidden staircase.



4

The Underworld

Armed with the flashlight, Harold walked easily down the secret staircase, which extended for some distance into a gray mist. A small, empty room with damp stone walls was at the bottom of the stairs. On the other side of the room there was a door with light shining from under it.

Harold opened the door and found himself in a sunny courtyard. He was confused because he thought he had to be underground. Turning around, Harold saw that he had just come out of an old stone house that looked like a country manor and nothing at all like the small white house he had entered a few minutes earlier.

The courtyard was surrounded by bushes. There were trees in the distance. In the middle of the courtyard a smiling fat man in a tuxedo stood behind a roulette wheel. The thin man Harold had met earlier was standing next to the fat man.

A large pile of gambling chips was in front of the fat man, who had just spun the roulette wheel and tossed the ball into the wheel in the opposite direction. The thin man made a bet by placing a chip on the betting area next to the wheel.

"Come on red!" shouted the thin man, as the wheel spun. The fat man just smiled. Harold got nervous when people smiled a lot. Finally the wheel came to rest.

"Seventeen—black. Sorry Frank," said the fat man who took Frank's chips from the table and didn't look particularly sorry at all.

"I'm down to my last seven chips," said Frank sadly. "My luck's got to change soon." Frank closed his eyes. "I can feel it!" he said convincingly. "I know I'm going to win this spin."

"Place your bets," said the fat man, as he prepared to spin the wheel. Frank placed two chips in the green section marked zero. The fat man spun the wheel.

"Come on, zero!" shouted Frank.

"Black twenty-four," said the fat man when the ball came to rest.

"I can't believe it," groaned Frank.

"I can believe it," said Harold.

"What are you doing here?! I thought I told you to leave," said Frank, whirling around to face Harold.

"You're bound to lose in repeated play," said Harold. "If you play roulette the way it's played in casinos," continued

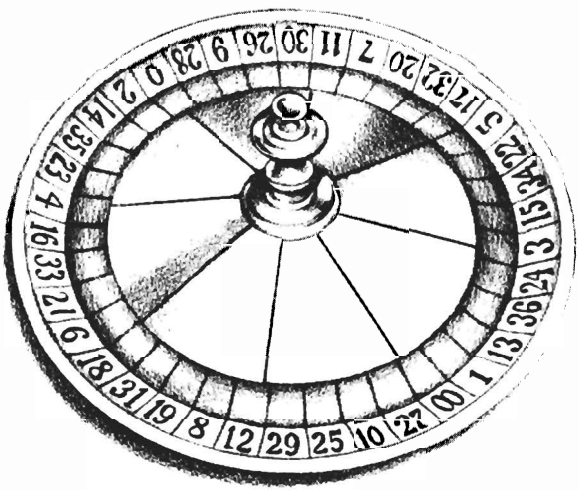
Harold, "the casino has the advantage no matter how you bet. You might be lucky some of the time, but you'll be unlucky more often. No matter how much you start with, you'll eventually go broke."

"Very interesting," said the fat man, who glared at Harold. "Give it another try, Frank," smiled the fat man. "Your luck's bound to change."

"I've got a hunch that I'll win now," said Frank as he placed a chip in the box marked "RED."

After Frank made his bet, the fat man spun the wheel.

Here is a program that simulates the casino game of roulette. A roulette wheel is divided into 38 sections numbered 1–36, 0, and 00. Half the numbers from 1 through 36 are colored red, the other half black, while 0 and 00 are green. In the gambling game, players place bets on colors, numbers, and combinations of numbers. Then the wheel is spun and a ball is tossed into it. When the wheel stops, the ball comes to rest in one of the sections and the winning bets are paid off.



First the program will print the result of simulated spins of the roulette wheel. Since there are 38 sections in the wheel, you use INT(38*RND(1)) + 1 to generate a random digit between 1 and 38. The colors of the sections do not follow a simple formula so they will be stored in DATA statements. For example, since the section numbered 3 is red, the third element in the DATA list is R. Reading R or B from the DATA statement will tell the computer whether the selected section is red or black. Numbers 37 and 38 denote 0 and 00, respectively, and are recognized as green sections.

Clear the computer's memory then enter and RUN the following program. The TAB statement in line 150 tells the computer to print A$ starting at space 4 going across the line.

```
 10 PRINT " SHIFT  CLR/HOME "
 20 PRINT "SPINS OF A ROULETTE WHEEL"
 30 PRINT
 50 X=INT(38*RND(1)) +1
 60 IF X=37 THEN 200
 70 IF X=38 THEN 250
 80 FOR J=1 TO X
 90 READ A$
100 NEXT J
110 IF A$="B" THEN A$="BLACK"
120 IF A$="R" THEN A$="RED"
150 PRINT X; TAB(4) ; A$
190 GOTO 300
200 PRINT "  0 GREEN"
240 GOTO 300
250 PRINT "  00 GREEN"
300 FOR D=1 TO 800: NEXT D
310 RESTORE
320 GOTO 50
500 DATA R, B, R, B, R, B, R, B, R, B, B, R, B, R
510 DATA B, R, B, R, R, B, R, B, R, B, R, B, R, B
520 DATA B, R, B, R, B, R, B, R
```

Now to modify the program to allow betting. The program will allow you to bet only on red, black, or an individual number. (There are other bets available in roulette, but we will not include them here.) Since there are 18 red sections, 18 black sections, and 2 green

sections on the wheel, the chance that red will come up is 18/38 = .47 and the chance that black will come up is 18/38 = .47. On the other hand, the chance that an individual number will come up is 1/38. Thus, if you bet on red or black you are much more likely to win than if you bet on an individual number. The casinos allow for this by varying the payoff odds. If you win a bet on red or black, the casino will pay 1 to 1 odds. (For every dollar you bet, you get a dollar in winnings.) If you win a bet on an individual number, the casino will pay 35 to 1 odds ($35 in winnings for every dollar bet).

Clear the computer's memory then enter and RUN the revised program. The *VAL statement* is used in line 420 to allow the computer to read the string variable B$ as if it were a numeric variable in order to compare it with the numeric variable X. If a line in the program is longer than the line on the screen (e.g., line 310), it will automatically continue on the next line on the screen. Don't press the **RETURN** key until you are finished typing the entire line in the program. The computer uses the **RETURN** key to recognize the end of a program line. If a program is longer than the screen, you can LIST part of it as follows: LIST 1-100. (This command lists lines 1-100.)

```
 10 F=500
 15 PRINT "SHIFT CLR/HOME"
 20 PRINT "WHAT IS YOUR NAME?"
 25 INPUT M$
 30 PRINT "GREETINGS, ";M$;"."
 35 PRINT "WELCOME TO FRIENDLY ROULETTE"
 40 PRINT "YOU HAVE $";F;" TO PLAY WITH"
 44 PRINT
 45 GOSUB 450
 50 X=INT(38*RND(1))+1
 80 FOR J=1 TO X
 90 READ A$
100 NEXT J
110 IF A$="B" THEN A$="BLACK"
120 IF A$= "R" THEN A$="RED"
130 IF A$="G" THEN A$="GREEN"
140 IF X=37 THEN X=0
145 IF X=38 THEN GOTO 250
150 PRINT X;TAB(4);A$
```

```
160 PRINT
240 GOTO 300
250 PRINT " 00 GREEN"
260 PRINT
300 FOR D=1 TO 800: NEXT D
310 IF B$⟨⟩"RED" THEN IF B$⟨⟩ "BLACK" THEN GOTO 400
315 IF B$⟨⟩A$ THEN GOTO 650
320 GOTO 615
400 IF B$="0" THEN IF X=37 THEN 600
410 IF B$="00" THEN IF X=38 THEN 600
420 IF VAL (B$) =X THEN GOTO 605
430 GOTO 650
450 PRINT
455 PRINT "WHAT WILL YOU BET ON, ";M$
460 PRINT "RED, BLACK, OR NUMBER?"
470 INPUT B$
475 PRINT
480 PRINT "HOW MUCH WILL YOU BET";
485 INPUT M
490 IF M⟩F THEN GOTO 530
495 PRINT
500 PRINT "OK, ";M$;" YOU BET $";M;
505 PRINT " ON ";B$
510 PRINT
515 PRINT
520 RETURN
530 PRINT "SORRY ";M$;", YOU ONLY "
535 PRINT "HAVE $";F
540 GOTO 480
605 M=35*M
610 PRINT
615 PRINT "YOU HAVE JUST WON $";M
620 F=F+M
625 PRINT "YOU NOW HAVE $";F
640 GOTO 700
650 PRINT "YOU LOSE!!"
655 F=F-M
657 PRINT
660 IF F⟩0 THEN GOTO 695
665 PRINT "I'M SORRY, ";M$;", BUT "
```

```
670 PRINT "YOU SEEM TO BE BROKE. "
675 GOTO 675
695 PRINT "YOU NOW HAVE $"; F
700 RESTORE
710 FOR D=1 TO 1500: NEXT D
750 GOTO 45
800 DATA R, B, R, B, R, B, R, B, R, B, B, R, B, R
810 DATA B, R, B, R, R, B, R, B, R, B, R, B, R, B
820 DATA B, R, B, R, B, R, B, R, G, G
```

You can use the law of averages to analyze roulette bets. Suppose you play roulette and bet one dollar on red over and over. In a large number of plays you will win your red bet a fraction of about 18/38 = .47 of the time and lose a fraction of about 20/38 = .53 of the time. This means that, on the average, in every 38 plays you will win about 18 times and lose about 20 times. Betting one dollar a play, this will give you an average loss of $20 - $18 = $2 in every 38 plays. Dividing this $2 by 38 gives you an average loss of $2/38 = $.053, or 5.3 cents per play. In other words, if you play repeatedly, betting one dollar on red every play, you will lose at the average rate of 5.3 cents per play, eventually going broke. This number expressed as a percentage (5.3%) is called the *casino advantage*. The law of averages assures that there is no way to avoid this. You may be lucky for a short time, but eventually you will go broke. This is true for all roulette bets. Frank wasn't just unlucky. The same thing will happen to anyone.

"Number 9—RED," said the fat man, as the ball came to rest.

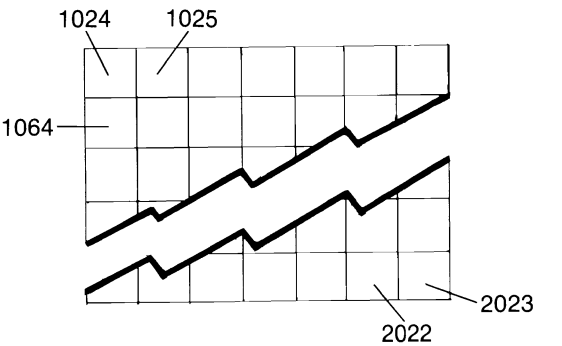
"Hooray!" shouted Frank. "I knew my luck would change!"

Frank won another bet but then lost the next three bets. A short time after that, he had lost all his chips.

"I can show you graphically what happens when you play roulette or any similar game," said Harold. "It's called a *random walk*." Harold took out his pocket computer.

A random walk is generated by a sequence of trials of a random experiment. After each trial, the next value of the random walk sequence is determined. The random walk can follow the fortune of a bettor who bets one unit on each play of a game in which there is a 50-50 chance of winning. Whenever the bettor wins, his fortune is increased by one unit, and the next point on the graph will be plotted one unit over and one unit up from the previous point. Whenever the bettor loses, his fortune decreases by one unit, and the next point on the graph will be plotted one unit over and one unit down from the previous point. The sequence will continue until the graph either leaves the top of the screen (bettor reaches his goal) or leaves the bottom of the screen (bettor is wiped out). If the graph reaches the right side of the screen without having reached either the top or the bottom, we will have it wrap around to the left side of the screen.

The Commodore 64 has a special section of its memory which contains information about screen graphics. The TV screen is divided into 25 rows with 40 character locations in each row (1,000 characters all together). A keyboard symbol like a letter or a number is one type of character. In order to get a specific graphics character to appear in a certain location on the screen you "POKE" its code into the appropriate location in screen memory. The 1000 screen positions begin in screen memory location 1024 and end in location 2023. Location 1024 is the upper left corner of the screen, location 1025 is one character to the right, and so on across the row. The second row begins in location 1024 + 40 = 1064. The second character in the second row is in location 1065, and so on. The lower right-hand corner is in location 2023. The illustration below demonstrates this.



In addition to the character memory, there are other memory locations that control colors on the screen. Location 53280 controls the border color and location 53281 controls the background color. There is also a separate color memory map that controls the color of characters on the screen (see the *Commodore 64 User's Guide*).

The codes of individual graphics characters, called "screen display codes," can be found in Appendix E in the *Commodore 64 User's Guide*. We will use a large dot to represent points on our graph. The dot has screen display code 81. For any screen location Z, the BASIC command POKE Z, 81 will poke the large dot into screen memory and thus plot it on the screen.

The Commodore 64 has colors that we can use for background, border, or character colors. Here is a list of the colors and their codes:

0 BLACK	8 ORANGE
1 WHITE	9 BROWN
3 CYAN	10 LIGHT RED
4 PURPLE	11 GRAY 1
5 GREEN	12 GRAY 2
6 BLUE	13 LIGHT GREEN
7 YELLOW	14 LIGHT BLUE
	15 GRAY 3

In the following program, the border and background colors are changed. You start the graph in the middle row at the left-hand side of the screen (location 1504). Proceed across the screen by using the location formula Z = 1504 + X + 40*Y, where X is the horizontal direction, increasing by one after each play, and Y is the vertical direction, causing the graph to go up one unit after each win (1 is added to the bettor's fortune) and down one unit after each loss (1 is subtracted from the bettor's fortune). Clear the computer's memory then enter and RUN the random walk program. In Commodore BASIC you can include more than one instruction on a line if you put a colon between each instruction (see line 70). To get back to the ordinary screen you will need to press the **RESTORE** key while the **RUN/STOP** key is pressed.

```
 20 PRINT "SHIFT CLR/HOME"
 25 POKE 53281,7
 30 POKE 53280,0
 50 P=INT(2*RND(1))
 60 IF P=0 THEN P=-1
 70 X=X+1:Y=Y+P
 80 Z=1504+X+40*Y
 90 IF Z<1024 THEN GOTO 500
100 IF Z>2023 THEN GOTO 500
110 POKE Z,81
120 FOR D=1 TO 50:NEXT D
130 GOTO 50
500 GOTO 500
```

Now to observe a random walk in which the bettor has less than a 50-50 chance of winning. In roulette, the chance of winning a bet on red is 18/38 = .47. This slight advantage to the casino will cause the random walk representing the gambler's fortune to drift down, making it much more likely that the graph will end by leaving the bottom of the screen than by leaving the top. Since RND(1) generates a random decimal fraction, every decimal fraction has the same chance of being selected. Thus, the chance that this number is less than .47 is .47 and the chance that it is greater than or equal to .47 is 1 − .47 = .53. We will use this fact to generate a random walk representing the fluctuating fortune of a gambler who repeatedly bets on red in roulette.

Modify the program with these lines and RUN the revised program.

```
 40 R=RND(1)
 50 IF R<.47 THEN P=-1
 60 IF R>=.47 THEN P=1
130 GOTO 40
```

Now try varying the win probabilities by changing lines 50 and 60 above. For example, if you want the bettor's chance of winning to be .60, change .47 to .60. It is interesting to note that, even if the casino has only a slight advantage, the player's chance of being successful in repeated play is greatly reduced and vice versa.

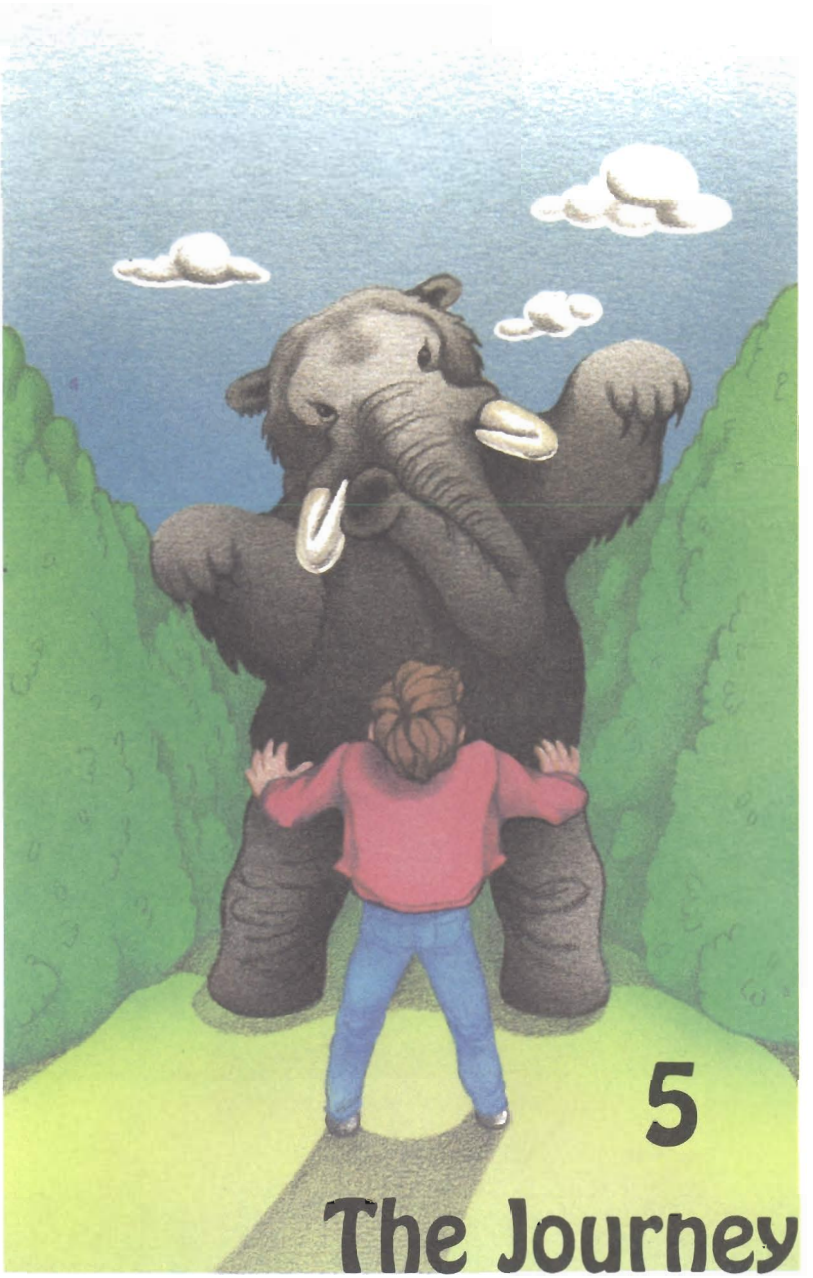
When Harold finished his demonstration, the fat man scowled.

"Very interesting," said Frank. "Show me some more."

"I really have to be going now," said Harold. "Could you tell me how to get to the maze?"

"First, leave the courtyard by crawling through the hole in the bushes over there," said Frank. "Then go through the tall trees until you get to the canyon. Then follow the river to the Cold Cave. Be careful."

Harold thanked Frank and said goodbye to him and the fat man. As he walked across the courtyard he heard Frank say, "Give me 10 chips on credit. My luck's bound to change." As he crawled through the hole in the bushes, Harold heard Frank yelling "Come on 32!"



5 The Journey

A large field was on the other side of the bushes. As Harold walked across the field he thought about the day's events. He had no idea where he was or how he got there. On the other hand, there was something familiar about this place.

Harold reached the end of the field and started walking along a path through a grove of tall trees. Birds chirped. A squirrel ran across the path. Harold would have enjoyed walking through these woods if he knew where he was. He was wondering why he had to know where he was in order to enjoy himself. Then he heard a strange noise.

At first Harold thought the noise was the sound of a cricket, and then he thought it was the sound of a car honking but, after hearing it again, he realized that it was an animal sound. Harold began to walk more quickly, but as he walked faster the sound only got louder. He was scared and didn't know what to do. The sound seemed to be coming randomly from all sides now. He started to run. The trees were denser and the path was getting narrower. Suddenly, with a loud crashing sound, the strangest animal Harold had ever seen jumped from the bushes into the path directly in front of him.

You can write a program to illustrate the woods. For trees use the graphics character that appears on the right side of the "A" key. These right-side characters are obtained by pressing the **SHIFT** key and the key you want at the same time. The characters on the left side of the keys are obtained by pressing the **C=** key and the key you want. As you can see by looking at your Commodore 64 keyboard, the keys that have graphics characters are the letter keys along with "+", "−", "@", "£", "*". Clear the computer's memory then enter and RUN the following program, which will display the woods. We will POKE changes in the border and background colors to give an indication of the strange colors that Harold encountered in the underworld. To get back the original screen colors, you will have to press the **RESTORE** key while the **RUN/STOP** key is pressed.

```
10 FOR I=1 to 999
20 PRINT " SHIFT A";
```

```
 30 NEXT I
 40 FOR B=0 TO 15
 50 FOR C=0 to 15
 60 POKE 53280,B
 70 POKE 53281,C
 80 FOR M=1 TO 1000:NEXT M
 90 NEXT C
100 NEXT B
110 GOTO 110
```

Now we will add the strange sound of the animal. In Commodore 64 BASIC, sound is determined by locations 54272 through 54296 in the computer's memory. To control the sound, appropriate values must be POKED into this part of the computer's memory. We will not describe the Commodore sound capabilities here. If you are interested in sound, refer to the *Commodore 64 User's Guide* or to the excellent book by Ed Bogas, *Make Your Commodore 64 Sing.*

The animal honks at random, so you write the sound part of the program in such a way that it is accessed at random. A part of a program that can be accessed as a unit from other parts of the program is called a *subroutine.* To branch to a subroutine, we use a *GOSUB statement,* which specifies the line number on which the subroutine begins. At the end of the subroutine, a *RETURN statement* sends the computer back to the program to the instruction just after the GOSUB statement. Subroutines are useful when a set of instructions has to be used repeatedly in a program.

You can access the sound subroutine at random by selecting a random number between 0 and 2 and using an IF/THEN statement that branches to the subroutine only if the random number is 1. Thus, the sound subroutine (lines 200–270) will be accessed about one-third of the times that line 75 is executed. The STEP variation of the FOR/NEXT loop in line 210 makes the variable X increment in units of 30. Now add these lines and RUN the revised program:

```
  2 POKE 54296,15
  4 POKE 54277,3
 72 R= INT(3*RND(1))
 75 IF R=1 THEN GOSUB 200
200 FOR Z=1 TO 5
210 FOR X=120 TO 150 STEP 30
```

```
220 POKE 54273,X-80:POKE 54272,X-80
230 POKE 54276,17
240 NEXT X
250 POKE 54276,16
260 NEXT Z
270 RETURN
```

The animal looked like it was part elephant and part bear. It had thick gray fur, large ears, and a long snout that almost touched the ground. The animal stared menacingly at Harold, pointed its snout at him, and let out a piercing shriek that sounded like Times Square at rush hour. Horrified, Harold turned and started to run back the way he came. He had gone only a few steps when the animal leaped over him and landed in front of him. It pointed its snout at Harold and honked loudly.

Harold didn't know what to do. Then he remembered the backpack. He opened the pack and offered the elephant-bear a handful of peanuts.

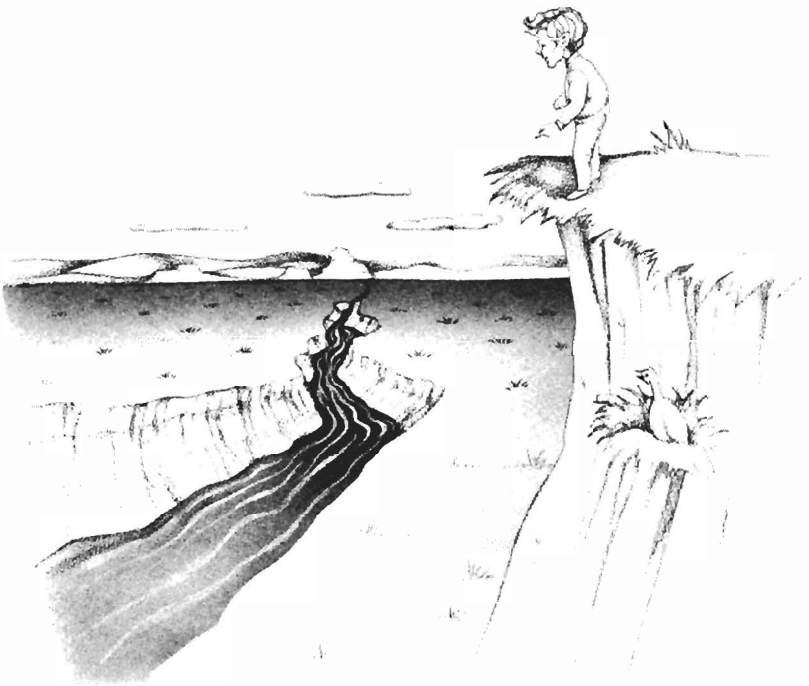
The elephant bear looked at the peanuts and honked again. Then it sucked the peanuts into its snout and spat them in its mouth. Harold offered the animal more peanuts, which it quickly ate. Before long it had eaten most of the peanuts. Then the elephant-bear gave Harold a friendly look and made a series of soft, beeping sounds.

"Well," said Harold to the elephant-bear, "it's been nice knowing you."

Harold turned and started to walk away. He had only gone a few steps when the animal caught up with him and put its warm snout affectionately in his hand.

"I know you want more peanuts but I only have a few left. Anyway, I have to find the maze." Harold sounded a lot more confident than he felt.

Harold removed the animal's snout from his hand and marched down the path. The elephant-bear paused for a moment and then bounded over to him. It wrapped its snout around Harold's arm and pulled him off the path into the woods.



"Stop!" screamed Harold, as he tried to break free, but the elephant-bear had a strong grip and wouldn't let go. It dragged Harold through a thicket of thorny bushes. It pulled him down a hill and across a stream. It yanked him through a beautiful field of orange and yellow flowers and along the top of a hill that bordered a grassy valley. Finally, it hauled the groaning boy up a rocky path to the top of a steep cliff which overlooked a large canyon. A fast river cut through the middle of the canyon and disappeared behind a rock formation at the canyon's northern end.

This was the canyon Frank had described. And there was the river that led to the Cold Cave. The boy patted the elephant-bear, which gave a contented beep.

The problem now was to get down the cliff. With the elephant-bear tagging close behind, Harold walked along the edge of the cliff but could find no path or even any ledges

that would have enabled him to climb to the bottom. Then he remembered the rope that was in his backpack. He got out the rope and tied it to a bush near the edge of the cliff.

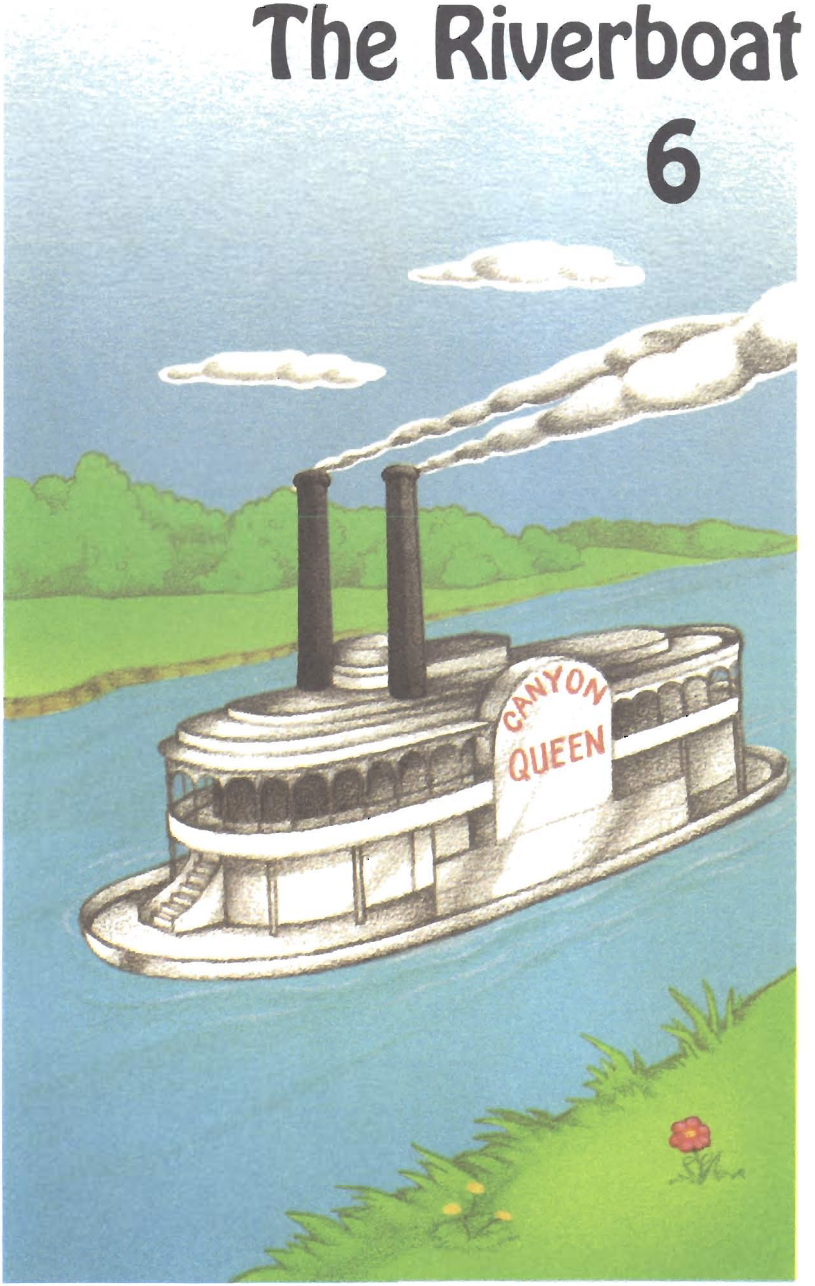
"Thanks for helping me, elephant-bear," said Harold as he shook the elephant-bear's snout. The elephant-bear honked and Harold swung over the edge.

When he reached the ground, Harold realized that the river he had seen from the top of the cliff was a good distance away. He started hiking across the canyon.

As he walked across a rocky field, Harold heard a rumbling sound in the distance. Suddenly, a man riding a mini-helicopter consisting mainly of a seat, an engine, and a propeller mounted on a shaft flew over the edge of the cliff and headed into the canyon. As the man flew past, Harold recognized Frank.

"Wait!" shouted Harold, but the thin man paid no attention as he roared out of sight above some tall trees.

The Riverboat 6



When Harold finally reached the river, his feet hurt and he had a blister on his ankle. The river was wider than it appeared from the cliff. Harold splashed water on his face and started walking along the riverbank toward the north end of the canyon.

As Harold walked along the bank the foliage got thicker until, finally, he could go no farther. He looked for an easy place to walk, but all he could see were trees and bushes. Harold sat down next to a large rock to rest. Then he stood up and stretched. Just for fun he took the toy horn from the pack and blew it, causing a shrill blast to echo across the river. Then Harold heard another sound—the low whine of a boat's engine. He looked up the river just as a large boat chugged into view.

The boat was a riverboat like the ones that traveled down the Mississippi River a hundred years earlier. The boat was in the middle of the river and traveling fast, for a riverboat. Harold could see people on the deck of the boat, but they didn't see him. He climbed up a rock and started blowing his horn.

Finally, someone on the deck of the boat heard the horn. Soon the boat changed course and came slowly across the river. A dinghy was dispatched to pick up the boy. Harold was now on the deck of a riverboat.

"Ahoy, mate," said the captain of the riverboat to Harold. "My name's Captain Salty and this here's the Canyon Queen." Captain Salty was a grizzled old man wearing a sailor's cap and smoking a pipe. A beautiful parrot sat on his shoulder. "We're going to the resort area at the north end of the canyon," said Captain Salty as he puffed on his pipe. "I'm glad we heard yer horn. Where ya headed?" Harold thought it was odd for a riverboat captain to have a name like "Salty" and started to wonder about the parrot, then gave up—today was not the day for things to make sense.

He told Captain Salty that he was trying to find the Cold Cave.

"That's not far from here," said Captain Salty. "I'll tell ya when we get there. Meanwhile, make yerself at home."

"Awk! At home! At home!" squawked the parrot. Captain Salty walked away, surrounded by a cloud of foul-smelling pipe smoke.

Harold wandered across the deck to where he saw some people were playing cards.

"I'll call 6 and raise you 3," said one player, a slender, serious, middle-aged woman who was smoking a thin, brown cigarette.

"I'll call your 3 and raise you10 more chips," said the other player, who was Frank! Harold waved and smiled at Frank, but the thin man was too busy playing cards to notice the boy. A young girl about Harold's age stood behind the woman.

"I've got three queens," said Frank.

"Three aces," said the woman, scooping up all the chips on the table.

"Good, Aunt Belle," said the girl, clapping her hands in delight.

"One more game," said Frank.

Harold watched, knowing what was going to happen. Sure enough, Frank lost.

"You cleaned me out," said the thin man.

"That's life," said Aunt Belle.

Here is a program that selects cards at random from an ordinary deck of cards. The card values and the suits are listed in DATA statements. The computer will first generate a random number between 1 and 13 to determine the card value (ace, two, three, . . . , king). It will then generate a random number between 1 and 4 to determine the suit (hearts, clubs, diamonds, spades). Combining these two things will yield a specific card. A FOR/NEXT loop picks the right items from the DATA list. Clear the computer's memory then enter and RUN the following program:

```
10 PRINT "SHIFT CLR/HOME"
20 PRINT "SELECTING A CARD AT RANDOM"
```

```
 30 PRINT
 35 PRINT
 40 PRINT
 45 X=INT(13*RND(1))+1
 50 Y=INT(4*RND(1))+1
130 FOR C=1 TO X
135 READ A$
140 NEXT C
142 RESTORE
145 FOR C=1 TO 13+Y
150 READ B$
155 NEXT C
160 RESTORE
165 PRINT A$;" OF ";B$
170 FOR D=1 TO 600:NEXT D
180 GOTO 45
500 DATA ACE,TWO,THREE,FOUR,FIVE,SIX
510 DATA SEVEN,EIGHT,NINE,TEN,JACK
520 DATA QUEEN,KING
530 DATA HEARTS,CLUBS,DIAMONDS,SPADES
```

In the game of poker, each player is dealt five cards. In other games, different numbers of cards make up a hand. You can modify the above program so that it selects a specified number of cards at random. Of course, in a card game you can't get the same card twice on one hand, so you must make sure that the computer doesn't pick the same card twice. You do this by using an array to remember each card that is selected. When a new card is selected, the computer will check through the cards selected previously to see if this card has already been selected. If so, the computer will select another card at random. It will keep doing this until the specified number of different cards are selected.

Add these lines to the program and RUN the revised card selection program:

```
 15 DIM Z(52)
 30 PRINT "HOW MANY CARDS (1–52)";
 35 INPUT W
 38 PRINT
 40 FOR K=1 TO W
```

```
 55 T=10*X+Y
 57 F=0
 60 FOR J=1 TO K
 65 IF T<>Z(J) THEN 80
 70 F=1
 75 J=K
 80 NEXT J
 85 IF F=1 THEN 45
 90 Z(K)=T
180 NEXT K
200 PRINT
210 GOTO 210
```

Frank got up from the table and walked over to his mini-helicopter, which was on the deck.

"I hear there's a craps game at the north end of the canyon," said Frank to no one in particular as he climbed onto his machine. "I'm lucky at dice." He turned the engine on and the propeller whirred above his head. Without saying goodbye, Frank flew off toward the north end of the canyon.

"Frank is a nice guy, but he can't stop gambling," said Harold to the girl after the thin man flew away. "He goes from game to game, losing every time; he can't seem to think about anything but betting. By the way, my name's Harold. What's yours?"

"Nancy," said the girl. "My Aunt and I are on vacation. Are you on vacation too?"

"Well, uh, sort of," said Harold, who tried to explain to Nancy how he had gotten there. This wasn't easy, as he wasn't quite sure himself.

"What an exciting adventure!" she exclaimed.

"It doesn't sound exciting to me," said Aunt Belle, who was putting the chips in her purse.

"Aunt Belle is a champion card player," said Nancy. "She specializes in poker and bridge. She doesn't like other games. Do you like video games?"

Harold started telling Nancy how much he liked video games and computers when, suddenly, Captain Salty appeared on the deck.

"Ahoy mates," said Captain Salty. "We'll soon be passing the jagged rock formation. At the end of the formation is a path through the rocks that leads to the Cold Cave. An old hermit lives in the caves. He's good people. He's been all over the world. Some say he can predict the future. I've been all over the world too. I think the future is random and can't be predicted."

"Awk! Can't be predicted!" squawked the parrot.

"Anyway, when we go by the jagged rocks," said Captain Salty, "we'll put you ashore in a rowboat. Good luck, kid."

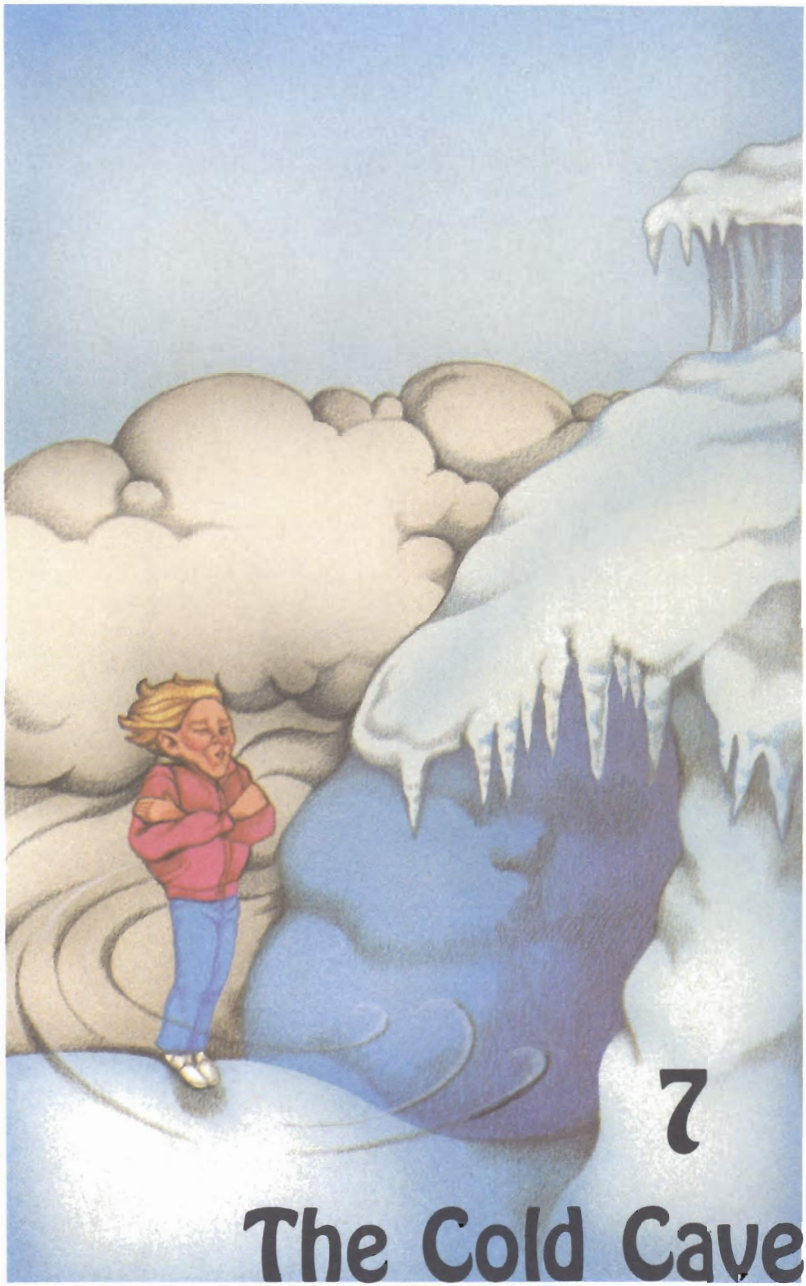
"Awk! Good luck!" squawked the parrot.

Captain Salty took a puff on his pipe and emitted a cloud of smoke that made Harold and the parrot cough violently.

Soon the jagged caves appeared. They were in a rock formation that went from the water's edge back to the canyon wall. Captain Salty prepared to lower the rowboat.

"It's been nice meeting you, Harold," said Nancy, shyly. "Here's a little present." Nancy blushed as she gave Harold a magazine entitled *Computer Madness.*

Harold thanked Nancy and put the magazine in his backpack. Soon he was on the shore.


7
The Cold Cave

The rocks along the shore formed interesting shapes, with small caves visible in various places. At one end of the formation, a path twisted its way into the rocks. Harold started walking along the path. It was late afternoon. As the sun began to set the air grew chilly and Harold started to shiver. After hiking for about 10 minutes Harold saw a large cave. A fog shrouded the cave and ice crystals adorned the rocks. "This must be the Cold Cave," realized Harold.

Harold walked to the cave entrance. A cold breeze stung his face and he wished that he had a jacket. Then he entered the cave.

Since the cave was dark, Harold took the flashlight from his backpack and turned it on. He was in a room with sleek rock walls and sparkling ice formations on the ceiling. On one wall was a sign that said **MAZE→**.

Harold walked in the direction of the arrow and found a narrow passage at the other side of the cave. Crouching down, he walked along the passage until he came to another room.

This room wasn't just cold, it was freezing. Harold began to shiver uncontrollably. He didn't want to turn back, but he couldn't go much farther without heat. There was a large rock in the center of the room. In front of the rock was something metal. Harold discovered to his delight that the object was a portable heater.

"Hooray!" shouted Harold. He rubbed his freezing hands together and then examined the heater. There was a small fuel tank and a tube to carry the fuel to the burner. There was a plunger to increase the air pressure.

Harold pushed the plunger and then realized that he had no matches. Looking around the cave, Harold found a pack of matches on top of a large rock. He held a lit match to the burner of the heater. Nothing happened. He repeated the procedure. Still nothing. He shook the heater. It was out of fuel. He looked around the room again. No fuel anywhere. If Frank had been there he would have told Harold that his luck had run out. Harold didn't really believe in luck running out, except for people like Frank who played gambling games that they were bound to lose eventually.

The boy shivered and then he remembered the green liquid in his backpack.

Harold pulled off his backpack so fast that he lost his balance and nearly fell over. He took out the jar of green liquid, opened it, and sniffed. It smelled like kerosene. He carefully poured it in the fuel tank of the heater. He lit a match, held it to the burner and, with a satisfying "poof," the fuel ignited.

Soon Harold was comfortable again. He picked up the heater and entered the next narrow passage.

After a long crawl, Harold reached another freezing room. This room was smaller than the first two. Harold could see no passages leading from this room except for the one he had just come through. An eerie light illuminated the room. In the center of the room was a large pile of furs. Harold laughed to himself. Wearing a fur or two would make him very cozy. This place wasn't so bad after all. He walked over to the pile and yanked at a warm looking fur.

"Watch it!" said the pile of furs.

The startled boy jumped back. "Wh-What?" Harold stuttered.

"The best philosophy is not to philosophize," said the pile of furs.

Harold tiptoed around the pile, looking carefully at the furs. "I must be getting tired," he muttered to himself.

"A journey of a thousand miles begins with a single step," said the pile of furs,

Harold looked more carefully at the furs, then pulled one from the top of the pile. This revealed what appeared to be a small person standing in the middle of the pile. "You're not a talking pile of furs," said Harold. "You're the hermit '"

"Things are not always as they appear. Conclusion: Don't jump to conclusions," said the pile of furs.

"Captain Salty told me about you," said Harold.

"If you pay too much attention to details, you forget about the important things in life," said the pile of furs.

"Could you stop philosophizing and tell me how to get out of here?" said Harold, who was getting annoyed.

"Question: Which is better, philosophy or nothing? Answer: Nothing, because nothing is better than philosophy!" The pile of furs shook with laughter.

"Very funny," said Harold, "but you didn't answer my question. You're just saying things at random."

"Random," said the pile of furs. "Random means uncertain. You can't predict the result. Around here, there's randomness in everything."

"You know what random means, so maybe you're not just saying things at random," said Harold. "Could you tell me how to get out of here?"

"Everything I say is false," said the pile of furs.

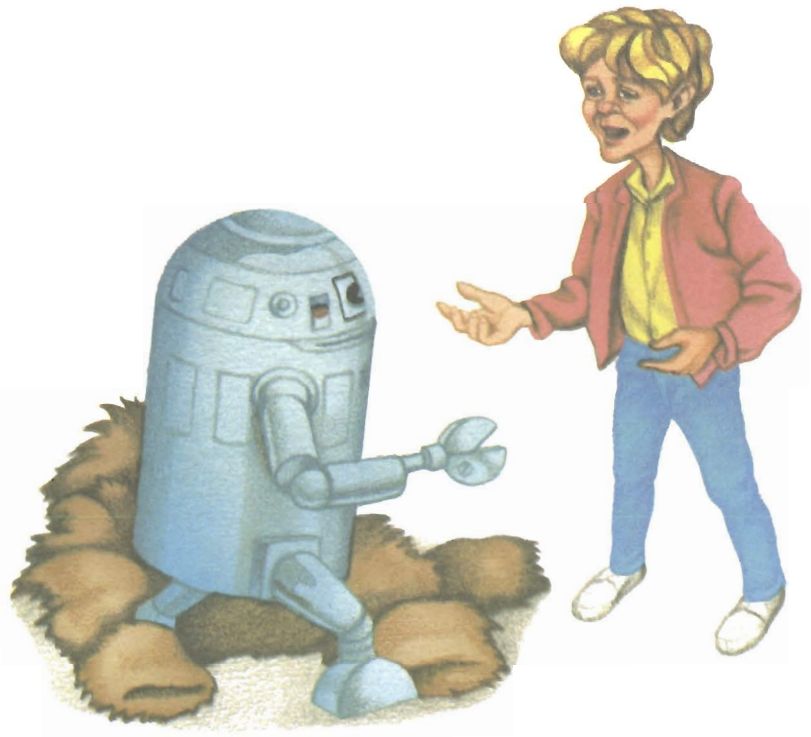
"All I want from you," said Harold angrily, "is the way out of here."

"Go back the way you came," said the pile of furs.

"This is ridiculous!" screamed Harold. He yanked another fur from the top of the pile, causing some more to fall off. Harold saw that under the pile of furs was—a robot!

"You're not a hermit . . . You're a robot!" shouted Harold.

"So what else is new?" said the robot.



A fun area of computer applications is "artificial intelligence," in which a computer is programmed to act like a person. In science, business, medicine, and other areas in which people need to use highly specialized or technical information, "expert" programs enable people to use a computer to access the knowledge of leading experts in the field. In other words, the computer acts as the expert and answers questions or supplies information according to the requests of the user.

The hermit in the Cold Cave is actually a robot programmed to act like a hermit. In addition to responding to certain words, the robot makes random comments and predictions.

The message you get in a fortune cookie is probably random. Anybody else could get the same message, and you could get any other message. But, maybe not. Maybe there are magic forces that direct fortune cookies to the right people.

You can write a program that makes comments at random and responds differently to yes or no answers to its questions. The program will be simpler than the program that controlled the robot-hermit. For one thing, you can't actually talk to a Commodore 64; you can only enter information at the keyboard. Programs that contain different responses to different words and which contain a lot of written data are necessarily long. Even to list ten comments from which to select at random requires ten lines of DATA statements. Subroutines will select a comment at random and respond a yes or no answer to a question.

Clear the computer's memory. Enter and RUN the following program. When you understand how the program works try to modify it so the computer will have the kind of personality you want it to have.

```
 10 PRINT "SHIFT CLR/HOME";
 25 PRINT "HELLO. WHAT'S YOUR NAME?"
 30 INPUT A$
 40 PRINT A$;"? THAT'S A GOOD ONE."
 50 PRINT "TELL ME, ";A$;", ARE YOU ";
 60 PRINT "HAPPY";
 70 INPUT B$
 80 IF B$="YES" THEN GOSUB 400
 85 IF B$="NO" THEN GOSUB 450
 90 PRINT "I THINK THAT"
 95 GOSUB 800
100 PRINT A$; " DO YOU LIKE THE WAY THINGS"
105 PRINT "ARE GOING"
110 INPUT B$
115 IF B$="YES" THEN GOSUB 400
120 IF B$="NO" THEN GOSUB 450
125 PRINT "IT'S FUNNY THAT"
130 GOSUB 800
155 PRINT "HOW OLD ARE YOU,"; A$;
160 INPUT B$
165 PRINT "REALLY? YOU DON'T LOOK IT."
170 PRINT "BY THE WAY,"; A$
180 GOSUB 800
185 PRINT "SO TELL ME," ;A$; " WHY ARE YOU"
190 PRINT "ALWAYS IN A BAD MOOD?"
```

```
195 INPUT B$
200 PRINT "THAT'S PRETTY WEIRD. I NEVER"
205 PRINT "WOULD HAVE THOUGHT THAT. "
210 PRINT "ACTUALLY, "
215 GOSUB 800
220 PRINT A$; ", DO YOU THINK THAT"
225 PRINT "ROBOTS ARE HERE TO STAY"
230 INPUT B$
235 IF B$="YES" THEN GOSUB 400
240 IF B$="NO" THEN GOSUB 450
245 PRINT "I'LL BET THAT"
250 GOSUB 800
255 PRINT A$; ", WHAT DO YOU THINK WOULD"
260 PRINT "MAKE YOU HAPPIER?"
265 INPUT B$
270 PRINT "THAT'S A LAUGH!"
275 PRINT "I'LL BET YOU DIDN'T KNOW THAT"
280 GOSUB 800
285 PRINT "DON'T YOU THINK THAT YOU SHOULD"
290 PRINT "GET MORE SLEEP?"
295 INPUT B$
300 IF B$="YES" THEN GOSUB 350
305 IF B$= "NO" THEN GOSUB 370
310 PRINT "ACTUALLY, I'M GETTING TIRED"
315 PRINT "JUST REMEMBER, "; A$; ", "
320 GOSUB 800
330 END
350 PRINT "I THINK SO TOO, "; A$
355 RETURN
370 PRINT "I DON'T BELIEVE THAT, "; A$
375 RETURN
400 PRINT "I'M GLAD TO HEAR THAT, " ; A$
410 RETURN
450 PRINT "I'M SORRY TO HEAR THAT, " ; A$
460 RETURN
500 DATA YOU WILL BE RICH AND FAMOUS
510 DATA PEOPLE THINK YOU'RE WEIRD
520 DATA YOU ARE BEING FOLLOWED
530 DATA YOU ARE FROM ANOTHER PLANET
540 DATA THIS SENTENCE IS FALSE
```

```
550 DATA NOTHING IS RANDOM
560 DATA YOU AGREE TOO EASILY
570 DATA EVERYTHING IS RANDOM
580 DATA SOMEDAY YOU WILL BE OLD
590 DATA WORK SHOULD BE FUN
800 X=INT (10*RND(1))+1
810 FOR I=1 TO X
820 READ D$
830 NEXT I
840 RESTORE
850 PRINT D$
860 RETURN
```

Harold thought about the robot. It seemed to be programmed to recognize certain words. If it didn't recognize any words in what you said, it would make a random comment. Harold was sure the robot was programmed to tell him how to get to the maze if he said the right thing.

"Okay, you electronic fortune cookie," said Harold. "How do I get to the *maze*?"

"To err is human, but to really screw things up it takes a computer," said the robot.

"I want to *escape*," said Harold.

"As long as you're happy," responded the robot.

"I want to go *home*," said Harold.

"I wish you would," said the robot.

"I want to find the maze so I can get back to a bus stop like the one near the park," said Harold.

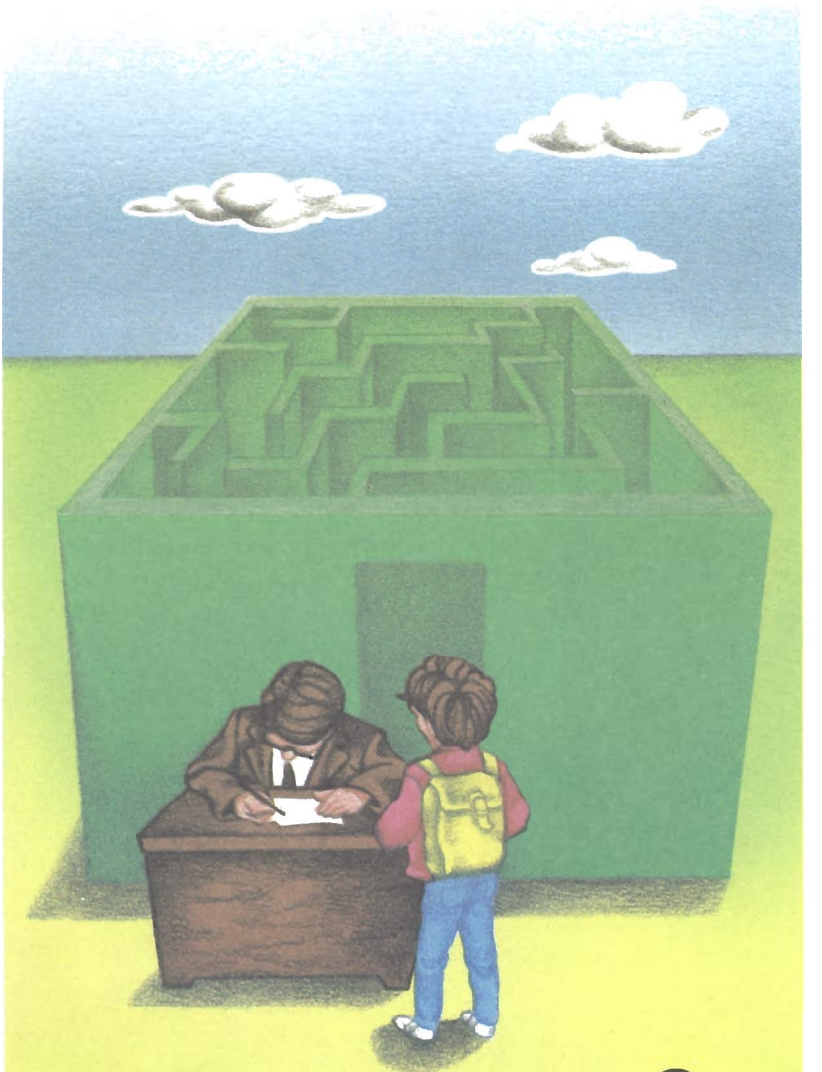
"Park?" repeated the robot. "To get to the park you must go through the maze. There is a small hole in the ceiling above my head. If you stand on me you can reach it."

"Hooray!" shouted Harold.

"Don't have a stroke," said the robot.

Harold climbed on the robot and pulled himself through a small hole in the ceiling. After a short crawl Harold came to

a wide passage. There was light at the end of the passage and soon the boy emerged into a flat, grassy field that was lit up by neon lights. There were tall rocks on all sides. At the end of the grassy field was a large sign that said ENTRANCE TO MAZE. TAKE A CHANCE.



8

The Maze

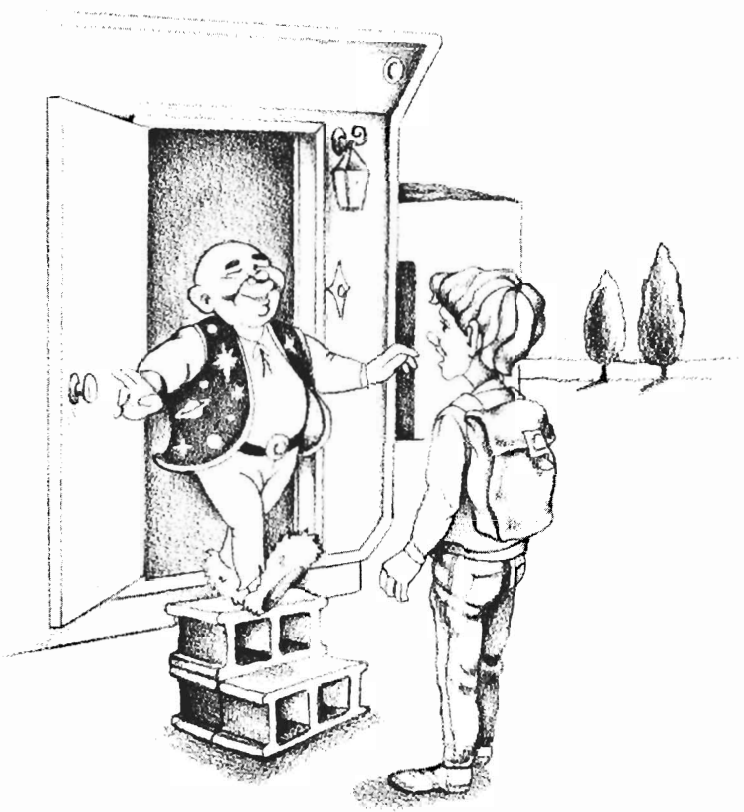
The maze was constructed of tall, carefully trimmed hedges. A man in a brown suit sat at a desk at the entrance to the maze, leafing through a pile of papers. A mobile home was parked next to the maze. As Harold approached the entrance, the man, without looking up, said, "Name, please."

"Harold Bloomgarden, but why do you want to know my name?" asked Harold.

"Rules," said the man, still leafing through his papers.

Harold couldn't understand why anyone would want to keep track of who went into the maze.

Harold was about to enter the maze when the door to the mobile home opened and a small bald man stepped out. The man was wearing bedroom slippers and a colorful vest embroidered with stars and planets.



"Well, what have we here?" said the man to Harold.

"We have Harold," said Harold nervously.

"Hello Harold," said the man. "I'm the Controller of Chance. I'm in charge of things here. Do you like games of chance?"

"Yes," said Harold. "In fact I was just about to enter the maze. It was nice meeting you."

"Everyone who lives here loves to gamble and play games of chance," said the man. "In fact, all we do here is play games. We have casino games, adventure games, board games, all kinds of games. Do you like to gamble?"

"Not really," said Harold. "I can't afford it. Also, I think it's boring."

"I'll show you a trick," said the Controller. He waved his arm and snapped his fingers. Suddenly a giant ball like the ones Harold had seen earlier came bouncing across the field. Harold leaped out of the way as the ball bounced past.

"I could have been crushed!" yelled Harold.

"So you could have," said the Controller, who was rocking with laughter. "I guess you were just lucky. Or maybe I planned it that way. I use Random Alley to get new people to come and play my games. Once you're here, it's not so easy to leave. That's a game in itself. You can enter the maze now. Good luck." The Controller laughed loudly. Then he turned and walked back into the mobile home. Harold ran into the maze.

You can write a program that prints a "random" maze on the screen by using the function CHR$(X), which prints a character on the screen whose ASCII code is X. The ASCII codes are used in various situations to provide numerical codes for the characters that can be typed on the screen. These codes and the corresponding characters can be found in Appendix F in the *Commodore 64 User's Guide*.

The characters in the program are the "slash" characters whose codes are 205 and 206. By picking one at random for each character location on the screen you can generate a maze that will look different every time. Although simpler in many respects, this

maze resembles the one Harold was in, including the colors, which you POKE into the background and border locations.

Clear the computer's memory. Then enter and RUN the following program:

```
10 PRINT "SHIFT CLR/HOME"
12 POKE 53280,7
14 POKE 53281,4
15 FOR K=1 TO 920
20 X=2*RND(1)
30 PRINT CHR$ (205+X);
40 NEXT K
50 GOTO 50
```

Harold found himself in a room walled by hedges. There were two other entrances to the room that led to other identical rooms. Harold walked into another room, then back to the first. Slowly he became aware of a loud ticking noise. Looking up, he saw a large timer on a platform suspended between two trees directly above the maze. The timer hand was currently at 3½ minutes. Harold wasn't sure what the timer was used for but he didn't like it.

Harold started racing from room to room. Meanwhile, the timer kept ticking. "Must be a Timex," Harold muttered. He ran into a room with no entrances other than the one through which he had come. Dead end.

Harold stopped to think. If he kept running around at random he would keep encountering the same dead ends. He backtracked from the dead-end room until he came to the first room with two other entrances. He reached into his backpack and took out the magazine that Nancy had given him. He tore out a page and attached it to the hedge next to the room that led to the dead end. If he returned to this room, he would know not to go that way. Then he left the room. The timer now showed 2 minutes.

Another sequence of rooms led to yet another dead end. Harold backtracked and marked the second dead-end passage with another page from the magazine. Then he went off

in a different direction. He was about to enter another room when a buzzer went off. The timer had run out.

Harold heard a rustling sound in the hedge behind him. Two burly security guards appeared. They grabbed him and dragged him through the maze.

"What's going on?" shouted Harold.

"Your time ran out," said one of the guards.

The guards dragged Harold back to the entrance of the maze, depositing him in front of the desk.

"Name, please," said the man in the slick suit, without looking up from his papers.

"Harold Bloomgarden," said Harold. "What's the meaning of this?"

"You get 4 minutes to go through the maze," yawned the man in the suit, putting a mark in his notebook next to Harold's name. "If you don't get through in time the guards bring you back. You get three tries all together. If you don't make it, you can't try again for a month."

"A month?!" said Harold. "But that's impossible! I don't even live around here!"

"I don't make the rules," shrugged the man.

Obviously Harold had no choice. He ran back into the maze. Using the same technique as before he began marking dead-end passages as quickly as he could. Once again, he kept finding dead ends. Finally, he ran into a different dead-end room. He found Frank and the fat man playing craps.

"Come on 6!" shouted Frank as he rolled the dice.

"Seven it is," said the fat man with a smile. "Sorry Frank," said the fat man as he raked in Frank's chips with a curved stick.

"Now I'll use the double-or-nothing strategy," said Frank. "You'll be interested in this," he said to Harold.

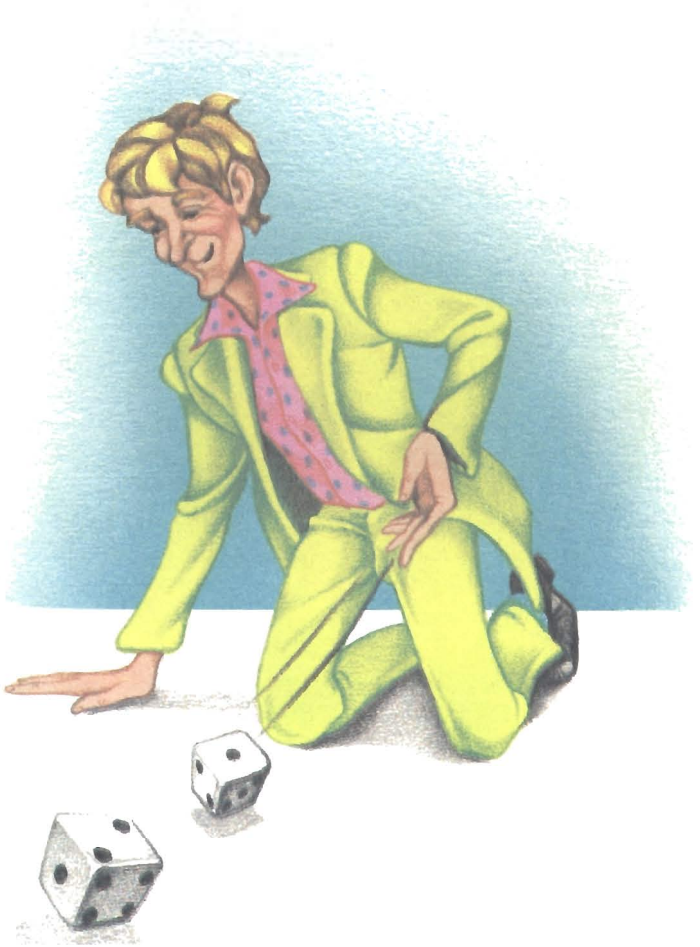
"First I'll bet one chip. If I win, I'll quit. If I lose, I'll bet two chips. If I win I'll have covered my first loss and I'll still be one chip ahead, so I'll quit. If I lose, I'll double my bet and bet four chips. If I win I'll have covered the $1 + 2 = 3$ chips I lost on my first two bets and still be one chip ahead, so I'll quit. If I lose I'll double my bet and bet eight chips. Eventually I'm bound to win. Then I'll quit!"

Harold watched as Frank rolled the dice. The dice came up 5. "Come on 5!" shouted Frank. He rolled the dice and 4 came up. After that 7 came up. "That's okay," said Frank, as the fat man took his chip, "I'll just double my bet." Frank bet two chips. After a short sequence of rolls, Frank lost again. "That's still okay," said Frank, as the smiling fat man took his two chips, "I'll double my bet again." Frank soon lost his four chips. "I lost one plus two plus four chips so far but now I'll double my bet and bet eight chips, which will cover my seven chips' loss and still let me come out ahead," said the confident Frank.

Before long, Frank had lost his eight chip bet as well as a 16 chip bet, a 32 chip bet and a 64 chip bet. Frank now had only seven chips left. "Loan me enough chips to double my

bet," said Frank to the fat man. The fat man laughed and loaned Frank the 121 chips necessary to make a 128 chip bet. The first roll of the dice was 11.

"Hooray!' shouted Frank. "A big win! This covers my losses of 1, 2, 4, 8, 16, 32, and 64 chips and still leaves me . . . uh . . . 1 chip ahead . . . Anyway, now I can quit a winner." Frank was



upset because he hadn't realized that no matter how large his bet in the double-or-nothing sequence, when he won he would only come out one chip ahead.

As Harold left the room, he heard Frank say, "Maybe I won't quit after all. I don't like the double-or-nothing strategy because I hate to quit, especially if I've only won one chip. Let's play some more!"

Now for a program that simulates the casino game of craps. Craps is a dice game in which one of the players rolls a pair of dice repeatedly until certain results occur. The player rolling the dice is called the "shooter." Here is how the game is played.

If the result of the first roll (the sum of the dots on the upturned faces) is 7 or 11, the game is over and the shooter wins. If the result of the first roll is 2, 3, or 12, the game is over and the shooter loses. If the result of the first roll is anything else (4, 5, 6, 8, 9, 10) the outcome is called the "point." The shooter then keeps rolling the dice until either the point is matched or 7 comes up. If the point is matched before 7 comes up, the shooter wins. If 7 comes up before the point is matched, the shooter loses.

For example, if the result of the first roll is 8, the shooter keeps rolling the dice until 8 or 7 come up. If 8 comes up before 7 the shooter matches the point and wins. If 7 comes up before 8 the shooter loses. After the first roll only the point and 7 have any significance. Every other result is ignored, including 2, 3, 11, and 12, which cause the game to end if they occur on the first roll.

In the craps program, the computer will print the results of each roll and whether or not the shooter wins. In order to let the computer know whether or not a particular roll is the first roll, you can use what is known as a flag. A *flag* is an indicator that lets the computer know if a certain condition is met. The flag will be the variable R. R will have value of 0 on the first roll. After the first roll is made, we will set R equal to 1. On every roll the computer will check the value of R to tell which stage the game is in.

Clear the computer's memory then enter and RUN the following program:

```
 10 PRINT "SHIFT CLR/HOME"
 20 PRINT "THE GAME OF CRAPS"
 25 PRINT
 27 PRINT
 30 R=0
 50 X=INT (6*RND(1)) +1
 60 Y=INT(6*RND(1)) +1
 70 Z=X+Y
 75 FOR D=1 TO 400: NEXT D
 80 IF R<>0 THEN 160
 90 IF Z=2 THEN 115
100 IF Z=3 THEN 115
110 IF Z=12 THEN 115
112 GOTO 120
115 PRINT Z; "CAME UP! ";
117 GOTO 200
120 IF Z=7 THEN 135
130 IF Z=11 THEN 135
132 GOTO 140
135 PRINT Z; "CAME UP! ";: GOTO 250
140 PRINT "THE POINT IS ";Z
142 PRINT
145 P=Z
150 R=1: GOTO 50
160 PRINT "THE ROLL IS ";Z
170 PRINT
180 IF Z=7 THEN 200
185 IF Z=P THEN 250
190 GOTO 50
200 PRINT "THE SHOOTER LOSES"
210 GOTO 300
250 PRINT "THE SHOOTER WINS"
300 PRINT
310 PRINT "PLAY AGAIN (Y OR N) ";
320 INPUT A$
330 IF A$="Y" THEN 10: IF A$="YES" THEN 10
340 PRINT
```

```
345 PRINT
350 PRINT "THANKS FOR THE GAME!"
360 GOTO 360
```

When craps was first introduced in casinos, the only bet you could make was that the shooter would win. The chance that the shooter wins is .493, a little less than 50-50. The casino pays 1 to 1 odds on this bet, which would be fair odds if there were a 50-50 chance of winning. As it is, the casino has a slight advantage of 1.4% (better than the 5.3% casino advantage in roulette), but even a slight casino advantage insures that the persistent bettor will eventually go broke.

The following program will simulate craps for as many games as you want to play, keeping track only of whether the shooter wins or loses. The computer won't announce the results of each game, but will print the fraction of wins and losses at the end. In a large number of plays, the fraction of wins should be close to .493, the probability that the shooter wins. Clear the computer's memory then enter and RUN the following program:

```
 10 PRINT "SHIFT CLR/HOME"
 20 PRINT "CRAPS SIMULATION"
 22 PRINT
 25 PRINT "HOW MANY GAMES";
 27 INPUT Q
 29 PRINT
 30 PRINT
 32 PRINT
 35 PRINT "WINS", "LOSSES"
 40 FOR J=1 TO Q
 45 R=0
 50 X=INT(6*RND(1))+1
 60 Y=INT(6*RND(1))+1
 70 Z=X+Y
 80 IF R<>0 THEN 180
 90 IF Z=2 THEN 200
100 IF Z=3 THEN 200
110 IF Z=12 THEN 200
120 IF Z=7 THEN 250
130 IF Z=11 THEN 250
```

```
145 P=Z
150 R=1:GOTO 50
180 IF Z=7 THEN 200
185 IF Z=P THEN 250
190 GOTO 50
200 L=L+1
240 GOTO 260
250 W=W+1
260 PRINT W, L
270 PRINT "CRSR ⇑";
300 NEXT J
310 PRINT
320 PRINT
345 PRINT "FRACTION OF WINS = "; W/Q
350 PRINT
360 PRINT "FRACTION OF LOSSES ="; L/Q
370 PRINT
380 PRINT
```

The most popular casino games are craps, roulette, keno, slot machines, and blackjack. Except for blackjack, all these games have a casino advantage that is impossible for the bettor to overcome. (No matter what the betting system, in the long run the gambler will lose.) In the late 1950s mathematicians did computer simulations of the card game blackjack. By looking at millions of simulated blackjack games, they were able to devise strategies that would give the player an advantage over the casino in certain situations. These strategies were difficult to learn and involved remembering things about the cards as they were dealt from the deck. Even under optimal conditions, it is difficult for a skilled blackjack player to come out a winner.

As Harold ran down another passage, he heard a strange squealing sound. A group of wild pigs came running through the passage. Harold dove into a hedge as the pigs ran by. Moments later he climbed shakily out of the hedge. The timer read one minute.

Harold kept running through the maze marking dead-end passages as he found them. He was beginning to feel that he had been in this situation before, but he didn't know when. Then the timer ran out.

Harold was once again dumped unceremoniously in front of the desk at the entrance to the maze.

"Last chance," said the maze keeper, without looking up.

Harold ran back into the maze. This time he was confronted by three angry geese. The geese chased him into a low tunnel that he had to crawl through, and then he slid down a slippery incline. He ran across a narrow bridge stretched over a stream that seemed to run through the maze and he jumped over a large log that blocked his path.

Just when he thought he might be making progress, Harold ran into another dead-end room, only to find Frank and the fat man again. This time they were playing poker.

"Come and play with us," said Frank when he saw Harold.

"I can't," said Harold, pointing to the timer. "I don't have much time and I've got to get through the maze."

"I can turn off the timer," said the fat man with a smile. He took a small signal device from his pocket and pressed a button which caused the timer to stop. "Want some chips?"

Harold got some chips and began to play poker with Frank and the fat man. He won a few hands and lost a few hands, and then he was dealt four kings. Harold and Frank stayed in for a series of bets and the fat man dropped out. When they were finished betting, most of Harold's chips were in the pot.

"I've got four kings," said Harold, triumphantly displaying his hand.

Frank then showed his hand, which consisted of the two, five, and seven of hearts, the three of clubs, and the nine of spades. "I've got a lollapalooza'" shouted Frank. "I win!!"

"You've got a what?" asked Harold.

"A lollapalooza!" answered Frank. "A two, three, five, seven, and nine, or possibly an eight, not all of the same suit. A lollapalooza beats anything!"

With a smile, the fat man took the chips in the center of the table and gave them to Frank. "Want to play some more?" the fat man asked Harold.

"Okay," responded Harold. "A lollapalooza, my eye," he muttered to himself. "There's no such thing." After playing a few hands Harold had won four chips. Then he was dealt the two of diamonds, three of spades, five of spades, seven of clubs, and nine of hearts: a lollapalooza!

After a round of betting, there was a big pile of chips in the pot. The fat man dropped out of the betting, and Frank displayed his hand first.

"I've got three jacks," said Frank.

"And I've got a lollapalooza!" said Harold triumphantly. He reached out his hand to take the chips, but before he could scoop them in, the fat man had grabbed his wrist.

"Not so fast, pal," said the fat man. "Only one lollapalooza per day." Before Harold had a chance to reply, the fat man pushed the pile of chips over to Frank.

"That's not fair!" whined Harold. "You can't keep changing the rules."

"Why not?" said the fat man with a laugh.

"I guess there's not much I can do," reasoned Harold, but, suddenly he had an idea. "I am feeling lucky, though," added the boy as he took the list of registered voters from his backpack. "In fact," Harold continued, "I'm feeling so lucky that I'll bet you 10 chips that if you choose 30 people at random from this list, at least two of them will have the same birthday."

Frank took the list from Harold. "This list has thousands of names in it," he said. "Sure, I'll take that bet." Frank handed the list to the fat man who closed his eyes and pointed to a name.

"May 13," said the fat man as he read the birthday of the selected person. The fat man then chose another name. "September 10," he said. Continuing in this way the fat man chose 19 names and read the corresponding birthdays.

"September10," he said with a smile when he came to the 20th birthday. "That's a match."

"I can't believe it!" said Frank.

"I told you I was feeling lucky," said Harold, who gave all his chips to a surprised Frank. The fat man then restarted the timer. "Bye," said Harold.

Harold smiled to himself as he dashed from the room. He knew from studying the laws of chance that if you select 30 people at random, the chances are very high that at least two will have the same birthday.

You can write a program that simulates birthday selections. You will be able to input the number of "birthdays" you want selected at random. If 23 or more people are selected at random there is greater than a 50-50 chance that at least two have the same birthday. If you select 30 birthdays at random the chances are quite high that at least two are the same.

To simplify the program, each month will have 30 days. To select a month, the Computer will choose a random number between 1 and 12 and then read the appropriate month from a DATA list. A random number between 1 and 30 is then picked for the day. Instead of using months and days, we could just as easily pick a random number between 1 and 365, but it's more fun this way.

In order to keep track of the birthdays we will use a *two-dimensional array.* When a birthday is selected, the number of the month and the day will be compared with the previously selected birthdays to see if there is a match. If there isn't, these numbers will be stored as a pair of numbers in the array. Since each array element has a pair of numbers instead of a single number, we say that the array is two dimensional. The DIM statement in line 50, DIM A(X,2), reflects this, with X denoting the number of birthdays to be selected and 2 denoting the fact that two numbers (month and day) are stored for each birthday.

If there is a match, the computer will stop selecting birthdays and announce that there is a match. Otherwise, the random selections will continue. If the specified number of birthdays are selected and all are different, the computer will announce that there are no matches.

Clear the computer's memory. Then enter and RUN the birthday selection program.

```
 10 PRINT "SHIFT CLR/HOME"
 20 PRINT "BIRTHDAY SELECTIONS"
 25 PRINT
 30 PRINT "INPUT # OF BIRTHDAYS";
 40 INPUT X
 45 PRINT
 50 DIM A(X,2)
 60 FOR I=1 TO X
 65 Y=INT(12*RND(1))+1
 70 Z=INT(30*RND(1))+1
 72 FOR Q=1 TO Y
 75 READ A$
 80 NEXT Q
 82 RESTORE
 85 PRINT A$;Z
 87 FOR D=1 TO 400: NEXT D
 90 FOR J=1 TO I
 95 IF A(J,1)=Y THEN IF A(J,2)=Z THEN GOTO 110
100 GOTO 130
110 PRINT
112 PRINT "MATCH!!"
115 GOTO 115
130 NEXT J
135 A(I,1)=Y : A(I,2)=Z
140 NEXT I
145 PRINT
150 PRINT "NO MATCHES"
155 PRINT
160 IF X>24 THEN GOTO 180
170 PRINT "THAT'S NOT UNUSUAL"
175 GOTO 175
180 PRINT "THAT'S UNUSUAL!!"
185 GOTO 185
500 DATA JANUARY, FEBRUARY, MARCH
510 DATA APRIL, MAY, JUNE, JULY
520 DATA AUGUST, SEPTEMBER, OCTOBER
530 DATA NOVEMBER, DECEMBER
```

Harold had two minutes left. He thought he must be getting near the end because of all the obstacles in his way. Another dead end. Backtracking. One minute left. He ran down a dark passage and came into another dead-end room. Trying not to panic, he was about to retrace his steps when suddenly Frank came roaring overhead in his mini-helicopter. When he saw Harold, Frank flew down until he was hovering just above him. "Grab my hand' I'll get you out of here'" he shouted.

Harold reached up and grabbed Frank's hand. Frank pulled Harold up until he was able to hang precariously on the noisy machine. Then the mini-copter rose above the maze.

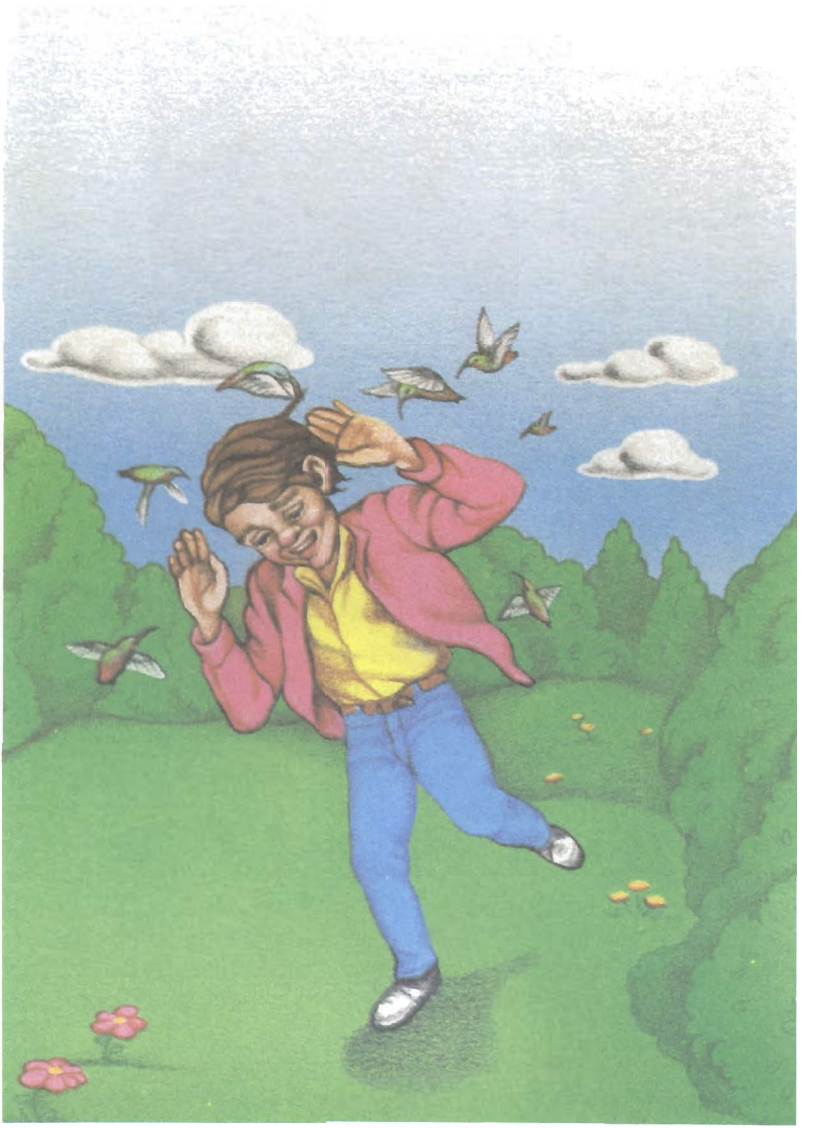
"There's no way out of the maze except the way you entered'" shouted Frank above the din of the engine. They were flying past the maze now. Harold could see the park below. The hedge in the park behind the little house was an outer wall of the maze. "Go back the way you came," said Frank, as he lowered the copter.

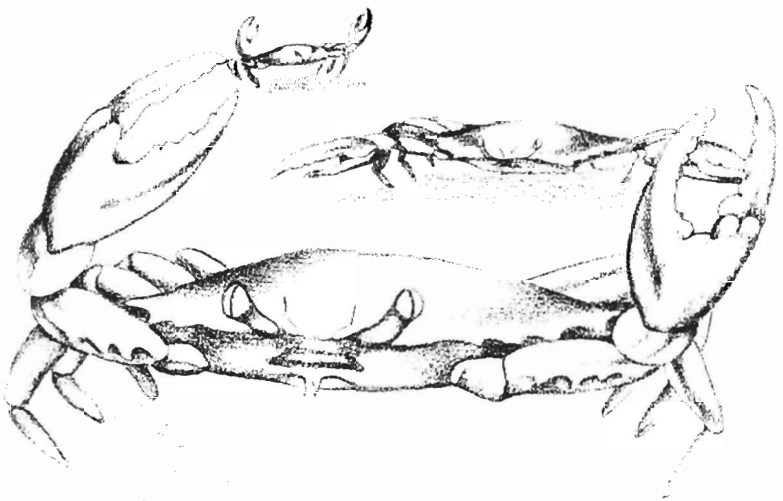
Harold jumped to the ground. He was in the park, a short distance from the little white house. "Thanks!" shouted Harold, as Frank flew away.

Harold ran back through the park as fast as he could. He was running across a grassy area when he heard a buzzing sound above him. He looked up just in time to see a formation of humming birds diving at him. They were making a screeching noise that made him sick. One of the birds jabbed him in the neck.

"Ouch!" shouted Harold, who waved at the humming birds with his hands, then suddenly remembered Louie's advice: Use branches to scare the birds. Harold ran over to a nearby bush but the branches were too strong for him to break off. He reached into his backpack and took out the knife. He cut off a branch and swatted at the humming birds as they came in for another attack. When the humming birds saw the branch, they broke formation and flew away. Harold threw down the branch and continued running across the park.

Harold slowed to a jog, trying to catch his breath when three giant crabs materialized from behind a tree and made straight for him. Harold ran faster, with the crabs in hot pursuit. He spotted the gray trees with the purple flowers that would





get rid of the crabs. He reached the trees and swung up out of the first crab's reach. He picked a flower, jumped out of the tree, and pointed it at the crabs. The horrified crabs crawled quickly away, and Harold continued running.

As he ran, Harold thought about the maze and the other events of the day. Even though he had left the maze he still felt like he was in a maze—or a game. Everybody here was playing games, and the games all seemed to be parts of a larger game. In fact, Harold felt like a character in a giant video game! That's what was familiar about this place! The frightened boy ran faster.

Harold reached the spot where Louie and Lena were playing the AIR-LAND invasion game. Harold waved to the elderly couple as he went running by. This time when he reached the street it was the same as when he had first come to the park. He stopped at the corner and took off the backpack. He took the change from the backpack and put it in his pocket. Then he tossed the pack onto a bench and left the park.

As Harold ran down the sidewalk he easily dodged some bouncing balls. They didn't frighten him anymore. He saw a bus heading uptown and got on it just as it left the stop.

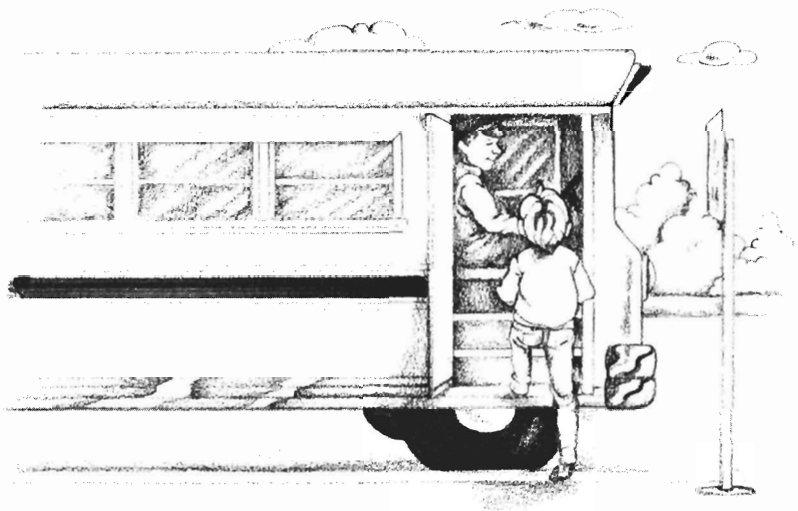
When Harold asked the driver how much the fare was, the change in his pocket was exactly the right amount. Nothing surprised Harold any more. The exhausted boy collapsed in a seat.

As the bus went across town Harold closed his eyes and relaxed. He thought about giant crabs and bouncing balls and mazes and humming birds and elephant-bears, about card games and dice games and adventure games.

Harold's thoughts blurred into dreams. He lived in a world of chance and games. He was in a maze and the maze was actually part of a larger game that was itself part of an even larger game, and so on. All of these games were part of one giant game that was run by the Controller of Chance.

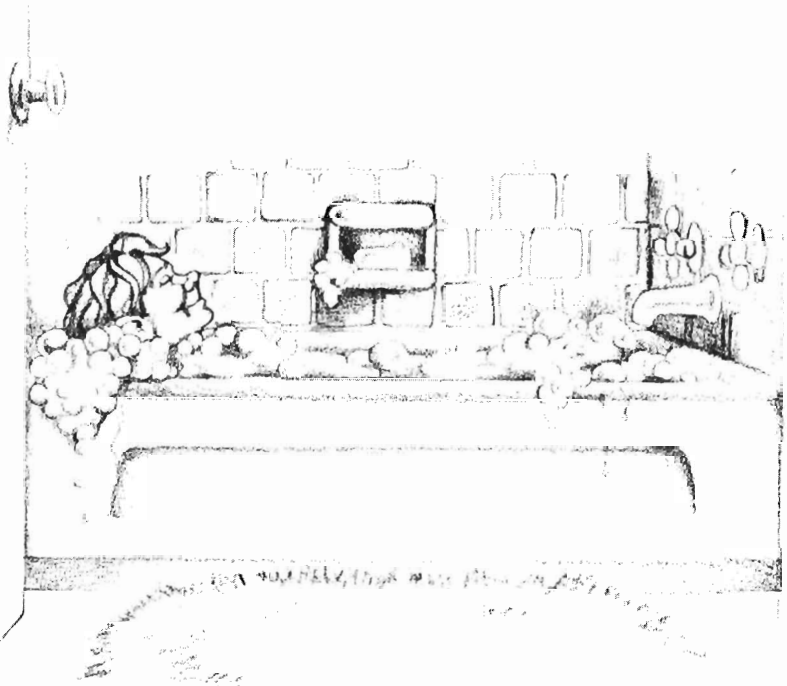
The next thing Harold remembered was rubbing his eyes and looking around as the bus went over a bump. The bus was on a street only a couple of blocks from Harold's house. It was night and the street lights were on. Harold walked to the front of the bus and asked the driver if they had gone through Random Alley.

"Random Alley? Never heard of it," said the bus driver. "Anyway, it's not on my route." Harold got off the bus and slowly walked the remaining block to his house.



"Harold Bloomgarden, where have you been?!" shouted Harold's mother when he walked in the door. "We've been looking all over for you. And look at you! You're covered with dirt and you have scratches on your face and arms! You take a bath right now!"

Harold thought about answering his Mother and telling her where he had been, but thought better of it. The odds were 20-1 she wouldn't have believed him.



$6.95

RANDOM ALLEY ADVENTURE FOR THE COMMODORE 64®

Michael Orkin

When Harold got on the bus he had no idea he would end up in Random Alley—a place where everything happens by chance.

You can follow Harold as he takes a walk through this fantastical world with your Commodore 64 home computer. You'll meet Frank, the thin man, who loses every game he plays; Captain Salty, captain of the riverboat Canyon Queen; the hermit in the Cold Cave; and many other colorful characters. As Harold encounters this animated collection of personalities, the RANDOM function is cleverly introduced. You'll learn how to flip a coin, throw dice, deal cards, and spin a roulette wheel—all on your computer. **RANDOM ALLEY ADVENTURE** gives you programs written in Commodore BASIC that you key in yourself, so you can see exactly what the program does and how it runs on the computer. You can even change the programs if you want to explore chance without Harold.

A Creative Pastimes Book
RESTON PUBLISHING COMPANY, INC.
A Prentice-Hall Company
Reston, Virginia

0-8359-6407-8